1996

Adept robot vision system and magnet-type end effector used for integrated circuit component assembly

Yu-Hung Hsu

Adept Robot Vision System and Magnet-Type End Effector Used for Integrated Circuit Component Assembly

by

Yu - Hung Hsu

A Thesis Submitted in Partial Fulfillment
of the Requirements for the
MASTER OF SCIENCE
in
Mechanical Engineering

Approved by:

Professor Wayne W. Walter
Thesis Advisor

Professor Mark H. Kempski
Dr. M. Kempski

Professor Michael P. Hennessey
Dr. M. Hennessey

Professor Charles W. Haines
Department Head

DEPARTMENT OF MECHANICAL ENGINEERING
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY

JUNE 1996

PERMISSION GRANTED:

I (Yu-Hung Hsu) hereby grant permission to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis entitled (*Adept Robot Vision System and Magnet-Type End Effector Used for Integrated Circuit Component Assembly*) in whole or in part. Any reproduction will not be for commercial use or profit.

July 10, 1996

Yu-Hung Hsu

ACKNOWLEDGMENT

Many people have helped me go through the past two years and have touched upon my life during this study. I would like to sincerely thank my advisor Dr. Wayne W. Walter for his guidance, understanding, friendship and environment conductive to learning, as well as for his contribution of expertise, time, patience and support. I would also like to take the opportunity to thank the members of my graduate committee, Dr. Kempski and Dr. Hennessey, for the hours they have spend reviewing my work, listening to my progress reports and reading my writings.

The final note of sincere thanks to my parents, sister, and Wen-Hsiu Lo, for their love, support, and encouragement.

Abstract

The machine vision system (MVS) has been heavily publicized because of its potentially crucial contribution to robotics. It is essential to alignment and inspection tasks in integrated circuit components' assembly equipment. The **objective** of this thesis was to examine the capability of a MVS to assist in the automated robot assembly of the dual-in-line device (DIL) and surface-mounted-device (SMD). The DIL, with 8×2-pin, and the SMD, with 24×16-pin, were used as models, while an *Adept SCARA* robot was used as the robot system. A magnet-type end effector, with a force sensor, was also designed to be incorporated with the vision system to measure applied force during assembly. A vision accuracy of 0.12mm/pixel and 0.12° was obtained, and an overall reliability of 91% for chip recognition and inspection was achieved. A complete assembly cycle time of 75 seconds, operating at a low robot speed of 50mm/sec (2in/sec), was recorded. The end effector detected assembly force with an average of 5.1% tolerance from the preset value. Polynomial fitting functions, calculated by using the Least Squares method, were employed to interpolate the experimental calibration data and to improve the force sensor accuracy of the end effector. Descriptions of the *Adept* robot system, vision system, and force sensing end effector are included.

Contents *Page*

Contents *Page*

List of Figures *Page*

List of Tables *Page*

CHAPTER I
INTRODUCTION

Today, the microprocessor is widely utilized in various fields. There are increasing demands that microprocessors be able to perform more powerful functions in the same sized package. In order to provide these powerful functions within the same space, more and more devices must be concentrated in a small area when fabricating a printed circuit board. Consequently, assembly and inspection of such a complicated array of components poses a critical problem. Most manufacturing processes require the parts to be properly positioned and oriented before being either introduced into a machine or handled by a placing machine. Position and orientation of a component sometimes can be preserved between operations; but this is not done as often as would be expected. Very few industrial robots in use today can perform "the job" when the components are not pre-positioned and pre-oriented. This situation causes product defects or reduces the reliability of assembly lines. Generally, there are three fundamental sources of error affecting integrated circuit component assembly: the robot placement system, the printed circuit board, and the component itself.

The first source of potential error is related to the robot placement system. A conventional robot placement system relies purely on mechanical structure to achieve its accuracy. Using such a machine requires operators to teach the robot every point on each new board. In addition, some transporting systems, for example the conveyor system, are

not able to stop at a desired location. Therefore, high accuracy is rarely obtained when using a traditional robot system; this significant source of error contributes to total placement error.

The second source of error originates from the *printed circuit board* (*PCB*). There are three types of error associated with the *PCB*. The primary concern is the fixturing of the *PCB*. Error can be introduced when the *PCB* is clamped in the fixture, due to the improper positioning or orientation of the *PCB*. Once the board is clamped, this error contributes to system inaccuracy. Printed materials on the *PCB* are a second concern. The printed materials must be properly positioned on specified locations for successful completion of future operations and processing. The third concern is due to the mechanical operations performed on the *PCB*. Operations such as drilling and milling, must also have proper positioning and orientation with respect to the circuit board. These errors cannot be overcome by current robot systems, because a robot will only execute a pre-taught program without any allowance for self-correction. Therefore, pin breakage will occur during assembly; this particular damage inhibits the achievement of a well-connected circuit.

The third significant error is due to defects in the chip itself. Manufacturing errors may cause chip leads to be distorted with respect to its body, or some pins may be missing. Since the chip body is used for the placement of the pins, any pin distortion will result in misalignment of pins to boards–even if the body is placed correctly. Although this component error can be eliminated by implementing a careful manufacturing process and/or establishing a pre-check step before the chips are loaded into the feeder, occasionally a

defective chip is introduced. This defective chip is referred to as a local error of the integrated circuit chip assembly process. To improve the accuracy of the system and eliminate the three fundamental sources of error, the use of industrial robots in this non-ideal environment requires sensors. When chips are not properly positioned and/or oriented, or boards are inaccurately fixtured, it may still be possible to employ an industrial robot equipped with suitable sensors to complete the required tasks. A suitable general-purpose sensor-based system is the *Machine Vision System* (*MVS*) [1-2].

Years ago, reports of successful *MVS* operations came mainly from universities and publicly supported research labs. In the last few years, a number of innovative systems have demonstrated the use of visual input and sophisticated computer processing to achieve a variety of manufacturing goals. Now, many of those early experiments are being converted to hardware and software which are utilized on factory floors [3-5]. Jim Field et al. has indicated *MVS* provides a cost-effective way of reliably placing surface-mounted-devices [6]. Precise mounting of imprecisely presented car bumpers has been proven feasible using a vision-guided robot on an automated assembly [7]. Also, the attachment of electrical connectors to a solar array was accomplished using a vision sensory system to perform the task [8]. Building on theses earlier examples, visual inspection has become a tool to assist humans with small parts sorting and inspection for manufacturing applications [9-10]. Visual inspection of chip placement is currently performed by human operators. In practice, this is a slow and costly process. Chip inspection is still an important visual task; thus there is an ever increasing demand for automatic optical inspection

systems to control chip assembly [11-12].

The first *MVS* application appeared in chip assembly in the late 1970s and early 1980s, replacing manual chip alignment with automatic systems. In 1975, the pilot *SIGHT-I* vision system was effectively demonstrated in the laboratory. It was fully integrated into production in 1977 [13]. The object of this *MVS* is to precisely handle a part and correctly insert it into a particular location.

Typically, a vision system consists of some fundamental equipment: a computer, a camera, a picture digitizer, special picture processing hardware, and governing software. Basically, this equipment is used to establish object images and present them on a computer monitor. These images are actually *2D* images; however, some theoretical *3D* images have been demonstrated by Chen and Vayda [14-15]. Acquired images are presented in two modes: binary scale images and gray scale images. Binary scale imaging is the easiest method for image display, because the object is seen as black or white by comparing its light intensities with a threshold value [16]. Gray scale imaging separates the light intensities into several different threshold values which can present images more accurately. After image acquisition, some vision operations may be applied during image processing. These vision operations analyze acquired images to establish the characteristics of the object. Therefore, the robot system can distinguish different objects or inspect some predetermined critical dimensions based on the gathered image information. To complete various operations, it is the gathered image information that is guiding the robot system to the correct location—with a proper orientation on the end effector. With the aid of a vision

system, the general errors described above can be eliminated, while also improving the accuracy of chip assembly.

This paper examines the efficiency of a *MVS* added to a robot system in order to assist in the tasks of chip acquisition, reorientation, and placement in a small scale chip assembly. The robot used for the assembly is an *Adept* robot system, equipped with a 'window-based' vision system. The images acquired by the vision system are presented as *2D* binary scale images on a monitor. In addition, various vision operations will be applied to these binary images in order to enhance chip recognition and inspection. Two different types of chips are used in the assembly: a *16*-pin *dual-in-line device* (*DIL*) and a *24×16*-pin *surface-mounted-device* (*SMD*). Considering the pin structure, it is reasonable to assume that the *DIL* is a *3D* object but the *SMD* is a *2D* object. Since the vision system can only analyze *2D* images, a special piece of equipment, i.e., a mirror, is employed to inspect the pin pitch and pin number of *DILs*. The lighting source used for the *DIL* is a small fluorescent light, which is mounted on the robot to provide uniform illumination. In the *SMD* case, pin width and pin pitch have smaller dimensions than the *DIL* does. Commonly, it is not easy to inspect such small dimensions using a projection lighting source, so a back lighting table is utilized to provide high contrast pin images for vision operation. On the other hand, the *PCB* is transported by a conveyor system which is controlled by digital I/O signals. A microswitch is used to generate these digital I/O signals; so the *PCB* can be transported to a pre-defined bounded area. In industrial chip assembly, the maximum acceptable force applied during the chip insertion is *0.05N* [17]. In order to achieve

this small load, a magnetic-type sensor-based end effector was designed and built to perform this task.

The contents of this paper include an exposition of the basic concepts of the hardware components and the *Adept* vision system in *Chapter II* and *Chapter III*. *Chapter IV* introduces the design algorithm of the end effector and its force calibration results. The assembly precision requirements are illustrated in *Chapter V*. The final discussions of overall assembly accuracy and vision performances are described in *Chapter VI*.

CHAPTER II

HARDWARE COMPONENTS

2.1 Adept Robot System

An *Adept* robot is used to perform the chip assembly processing. It has controlled path capability but paths and robot actions can be modified to suit the particular operation. Computer control and sensory feedback allow this *Adept* robot to react to its environment. The *Adept* robot is four-axis joint-type *SCARA* (Selective Compliance Assembly Robot Arm) robot; joints 1, 2, and 3 are rotational and joint 4 is translational. An angular servo control system is used to provide position and velocity feedback in order to position the joint motors to the desired angles [18-19]. In addition, the digital I/O board can be connected for signal input from sensors as well as output for controlling other equipment. The *Adept* robot is designed for direct drive, with a large work volume and a *9kg* (19.8lbm) payload. The work envelope of the *Adept* robot can be extended from *230mm* (9in) to *800mm* (31.5in), from the robot base, in the *X-Y* plane, and *470mm* (18.5in) in the *Z* direction, from top to bottom. This large work envelope and rotation capability provide a good working space and allow for the locating of objects in multiple locations. The *Adept* robot also incorporates the vision system as an important sensor. Based on image analysis, the vision system has the ability to guide the robot to a desired location. It is this vision system that makes the *Adept* robot overcome the drawbacks of traditional robot systems, and largely enhances its reliability [20]. The following figure (*Figure 1*) shows the configuration of the *Adept* robot and its work envelope.

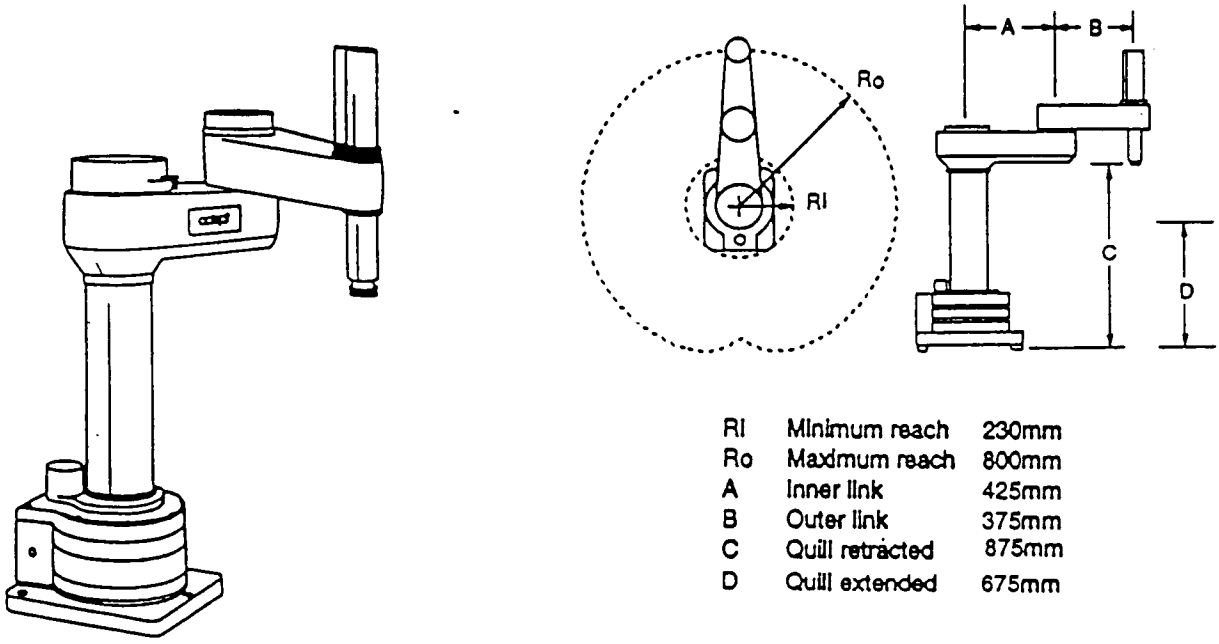


Figure 1. Configuration of the Adept Robot and Its Work Envelope

2.2 Chips and Printed Circuit Board

Integrated circuit components (*IC*) have been broadly utilized in computer systems. These components, developed from microprocessor technology, contain a number of bonding wires inside packages of different types and weights. The dual-in-line leaded chip (*DIL*), small outline transistor (*SOT*), small outline integrated circuit (*SOIC*, or simply *SO*), plastic leaded chip carrier (*PLCC*), and leadless ceramic chip carrier (*LCCC*) are commonly used [21]. Although there are many different types of chips, all of the chips can be briefly divided into two groups, based on their insertion methods: through-hole compo-

nents and surface-mounted components. A through-hole component is inserted into some aligned holes when it is mounted on the circuit board. Conventional solder technology is used to affix the chip into its location. Surface mounted components are innovative devices used instead of through-hole components. This type of components is simply placed on the circuit board with necessary footprints, so the lead width and lead pitch can be shrunk to a very small scale. In addition, through the aid of surface mounted technology, there is no requirement to drill or mill holes on a circuit board. Therefore, processing of surface mounted components is greatly reduced. Based on these two typical groups, a *dual-in-line component* (*DIL*) with *16*-pins and a *surface-mounted-component* (*SMD*) with *24×16*-pins were selected for this automated vision guided chip assembly project. The typical structures of these *ICs* are shown in *Figure 2.*

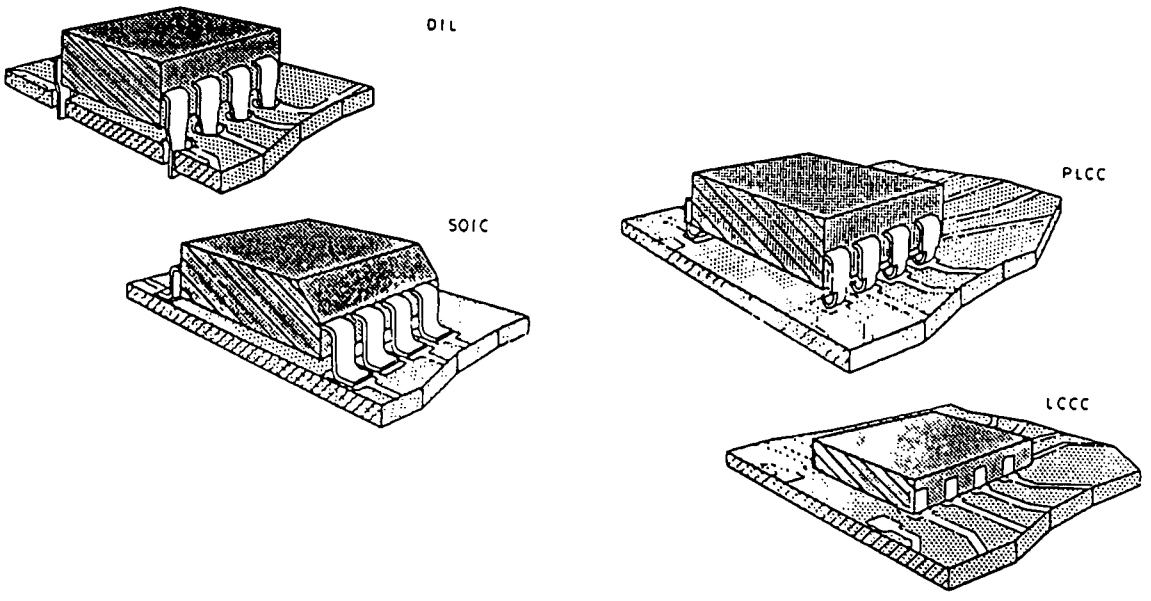


Figure 2. Chip Structures [21]

The *PCB* is utilized as the base on which to locate the integrated circuit components. It connects each component and constructs a well-defined circuit for chip communications and data transfer. There are always some aligned holes, printed marks, and footprints on the *PCB* to indicate the locations of circuit components. The custom *PCB* used for this project has been printed with necessary footprints and has drilled aligned holes for locating five *DILs* and one *SMD*. The dimension of the chip and the *PCB* used in this project are listed in *Table 1* and *Table 2*, respectively.

Type	Leads	Lead Pitch [p]	Lead Width [w]	Maximum Length [L]	Maximum Width of Body [A]	Maximum Width of Device [W]	Typical Height [C]
DIL	16	2.5	0.5	19	6	8	2.9
SMD	24×16	0.5	0.5	24	14	20	2

Table 1. Dimension of the DIL and SMD (mm)

Type	Maximum Length	Maximum Width	Lead Hole Diameter	Lead Hole Space	Footprint Number	Footprint Space
PCB	178	152				
DIL			1	1.5		
SMD					24×16	0.5

Table 2. Dimension of the PCB (mm)

2.3 Camera

Before a machine vision system can be applied to process an object, it is necessary to capture the object's image. This task is commonly undertaken by a camera. A camera

mounted on a robot system not only can be applied to capture the image, but can also be utilized to enhance the robot's ability. Since the objective of the robot system is to assist the user to perform faster and more accurate tasks, enhancing the ability of a conventional robot system also expands the capability range. In addition, by employing image analysis, a robot equipped with a camera can overcome some shifting and orientation errors of the object. Therefore, a traditional robot system can be enabled to process more complex tasks while acquiring the ability to increase its accuracy.

Since a camera is employed to provide the 'seeing' ability on a robot system, it is necessary to understand the structure and the function of a camera. Typically, cameras are composed of four basic parts. Each part performs a different task during image acquisitions, and is then combined with the others to engender a complete 'seeing' capability. These four parts are: the video camera body, the lens, the pixels, and the camera imaging surface.

Video Camera Body -- The video camera body is the camera itself without equipment, and is utilized for machine vision. It is generally used to construct a complete sequence for image acquisition and transformation. When the light enters the camera by passing through a lens mounted in front of the camera body, the camera body then supplies power to create charges for each part of the light sensitive electronic surface. After charges have been created and registered, the camera body will collect those charges corresponding to the light intensity entering the camera. These charges are digitized and transferred to the processor for image generation. After each complete operation, the power supplied to the

electronic surface will be temporally turned off and the camera body will wait for the next image acquisition command to acquire another image. It is this video camera body that establishes an image acquiring-sequence for image processing.

Camera Imaging Surface -- A camera imaging surface is used to acquire an image [22]. It is a compositional array of the pixel cells that store charges, replacing the film used in a traditional camera. A pixel, the smallest unit of information a vision system can return to the processor, is the basic unit of a picture element. The number of pixels the system can process determines the system resolution, which can affect the computer processing time needed to analyze an image. A pixel is a light sensor or photo detector working on an operating principle called 'charge coupling' [23]. In specific locations in the silicon semiconductor material, small amounts of electric charge are created by the field of a pair of gate electrodes situated very close to the surface of the silicon at that location. This charge generation is due to the release of free electrons in the semiconductor by light which has entered the camera body. When light falls on the pixel element, either in a row of individual elements or at the row-column intersection of an array, the initial charge is increased by an amount proportional to the light level. Once the charges have been gathered on the pixel, a photo detector placed at the end of each row (or column) in the pixel array counts the number of the charges and digitizes them. The digitized numbers inside an array indicate light intensity level received by the photo detectors. The value that is placed in that cell will be a shade of gray or a binary number: 0 or 1. Larger light intensities provide higher intensity values on the pixels; the color of the images acquired from those pixels are

lighter than the other. In most cases, these pixel values are presented as various gray levels, which are compared with a threshold value. If the values are smaller than the setting threshold value, such pixels are shown as black blocks. On the other hand, if values are equal or larger than the threshold value, then white blocks will appear. In order to re-present the image more accurately, it is sometimes possible to set two different threshold values. Once the second threshold is applied, the pixel values located in between the large and small threshold values are presented as white blocks. The pixel values higher than the large threshold value or lower than the small threshold value are seen as black blocks. A typical pixel array of a camera imaging surface is shown in *Figure 3*. The number inside each cell of the pixel matrix indicates the light intensity on that pixel.

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	61
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Figure 3. A Pixel Matrix and Light Intensities

When a single threshold value is applied on the pixel matrix, a pixel is shown either white or black by comparing the given pixel value with the specified threshold value; case 1. If a second threshold value, case 2, is applied, a pixel will be compared with both threshold values to present the correct pixel image. Sample values of case 1 and 2 are shown below in *Figure 4*.

Case 1

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	61
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Setting Threshold 1 = 30
Number higher than 30 ⇒ White
Number lower than 30 ⇒ Black

Case 2

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	61
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Setting Threshold 1 = 30
Setting Threshold 2 = 90
Number lower than 30 or
higher than 90 ⇒ Black
Number among 30 and 90
⇒ White

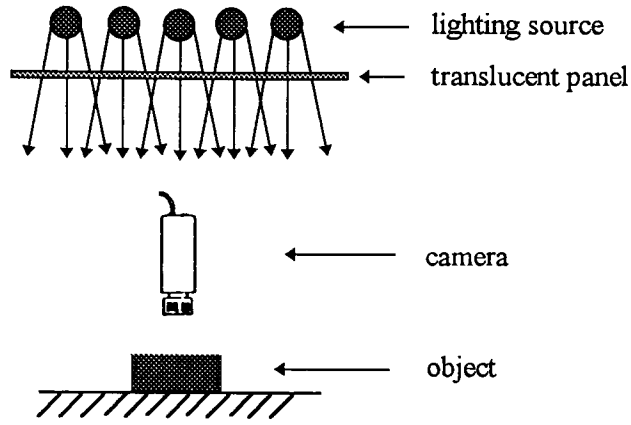
Figure 4. Threshold Setting Applied to the Pixel Matrix

2.4 Lighting Sources

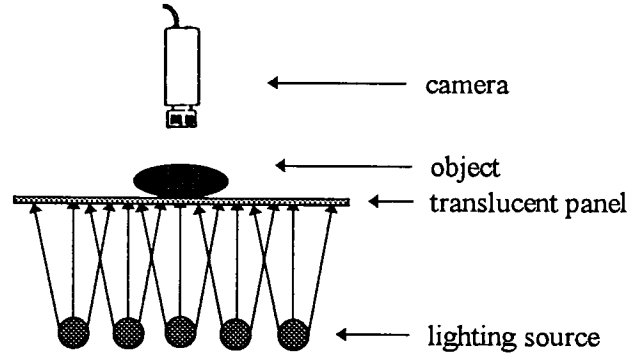
Because the vision system is totally dependent on the light reflected from the object to acquire an image, lighting is an important factor influencing a vision system, and the source and direction of lighting greatly effects the image representation. Typically, there are several different lighting sources which can be used with a vision system: diffuse lighting, back lighting, directional lighting, structured lighting, and strobe lighting [24]. *Diffuse lighting* illuminates a surface with light that strikes the surface from many different angles. Thus, shadows and reflections can be minimized. This lighting source can be applied on high-contrast objects, complicated objects, spherical objects, or objects requiring multiple inspections of interior features. A *back lighting* source is placed behind a part.

This type of lighting allows the vision system to ignore surface structure while concentrating strictly on the boundaries of a part, so it can be applied to a shiny part or a part which has a highly variable gray level on its surfaces. *Directional lighting* is placed at a special angle in order to illuminate some critical features of the object. Although this lighting source can be utilized to light a particular portion of an object, it also produces some unnecessary shadows. Those shadows will be seen as a part of the object. Thus, the real structure of an object will be misapprehend, and the vision operation will result in an incorrect analysis. It is therefore necessary to consider the mounting place, light direction, camera mounting place, and camera mounting direction to correctly capture an image. *Structured lighting* consists of a highly collimate light source to illuminate an object, for example, laser or lens focused light. It can reveal the length of a shadow or the depth of a surface feature. The shadow, in this case, becomes an important factor, which allows a vision system to inspect a special feature of an object. Commonly, some very small structures on the object are candidates for using the structured lighting. The last lighting source to be discussed is *strobe lighting*. Sharp, clear objects that are moving, often require strobe lighting (electronic flash). A lighting pulse of very short duration is generated to illuminate the moving object at a specific location. Most strobes are characterized by a very short duration of time that the light is on, often from *1/3000* of second to *1/10,000* of a second. The short duration will stop any movement, and the brightness will cancel any undesired light sources. This lighting source is primarily used when the object is carried by a conveyor. The above defined lighting sources are shown in *Figure 5*.

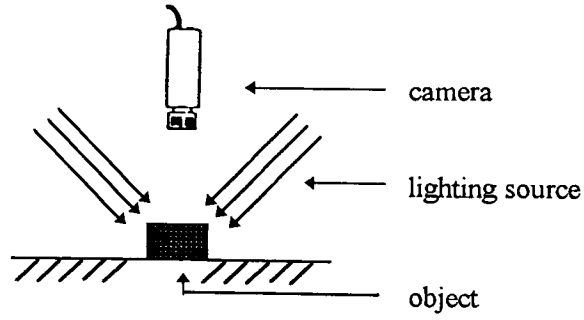
Diffuse Lighting



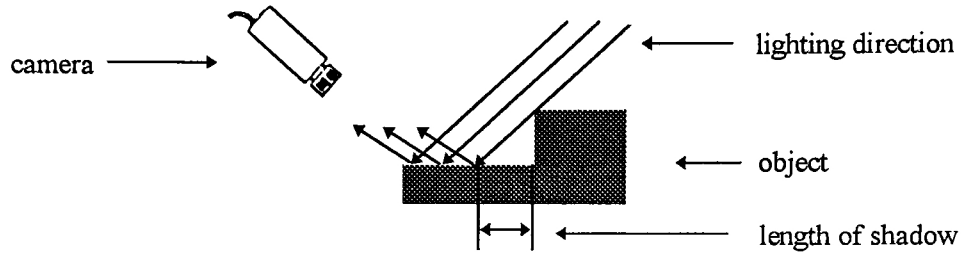
Back Lighting



Directional Lighting



Structure Lighting



Strobe Lighting

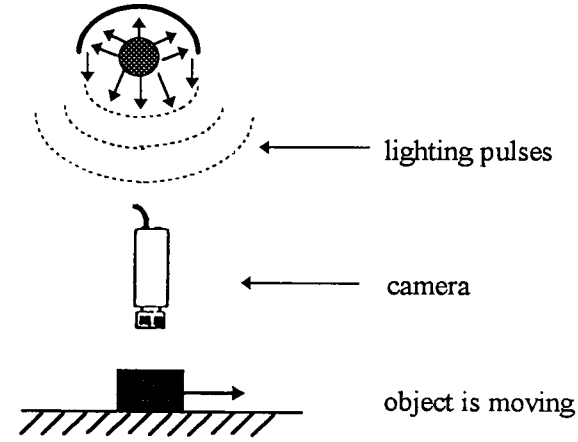


Figure 5. Various Lighting Sources

CHAPTER III

ADEPT ROBOT GOVERNING SOFTWARE

The system software that supports the operation of user application programs is an important feature of robot control systems. The software is not only used to control robot motion, but it also provides the capacity for the user to teach the robot as well. Although today's technologies have developed more powerful and functional software to support the increasing demand in regard to governing robot behavior, programming is generally a conventional method in most of the robot systems [25]. When a vision system is employed in conjunction with a robot system, programming technique will not be desired under such circumstances because it has no image processing abilities. Therefore, *Adept Technology Company* developed 'window-based' software to overcome the drawbacks of traditional programming methods. This 'window-based' software makes use of a computer mouse and a monitor to perform programming tasks. The user can visualize the object's images on a monitor and can simply modify those images through the 'click-and-drag' mouse function. Another benefit of using 'window-based' software control is that it provides easy use, reliability, and a flexible environment for robot control. There are four fundamental governing software programs inside the *Adept* robot system: assembly and information management (*AIM*) system, Visionware, Motionware, and the Sequence Editor. *Figure 6* shows the basic structure of this governing software.

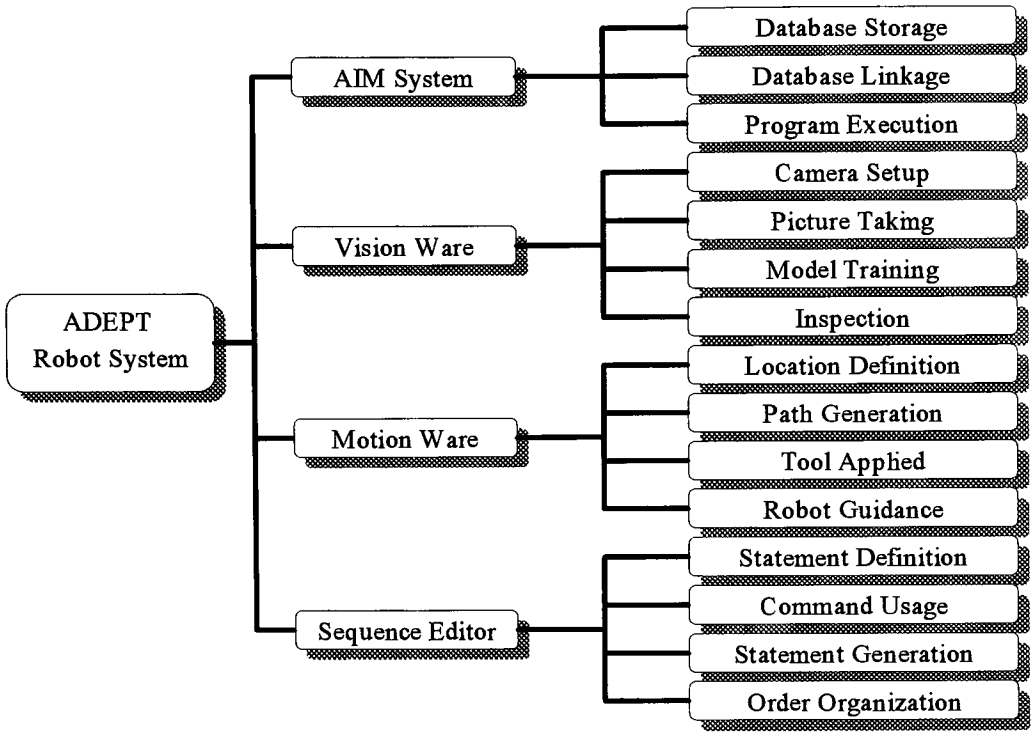


Figure 6. Basic Governing Software of Adept Robot System

3.1 Assembly and Information Management System

Assembly and Information Management (*AIM*) system is the *Adept* robot's top-level database management system. The database in the *AIM* system is used to keep track of all the data for the whole robot system. Within the database is a collection of records. These records are a key ingredient in the *AIM* system because they include the fields that hold individual pieces of information. Each portion of information is generated from other sub-software, such as Visionware or Motionware. After data analysis is completed, the *AIM* system will gather the acquired information, based on the information sources, and store them in a different databases. This step is referred to as database storage. Databases

commonly are divided into two groups: vision database and motion database. It is these two database groups that allow the *AIM* system to easily search the required information for further processing. After the databases have been established: when the program is executed, the *AIM* system can perform a task on the database: checking and linking. The *AIM* system first checks the gathered information (or value) in each record by comparing it with the value that has been defined by the user, for example, the length of an object's edge, the radius or a circle, and the distance between two points, etc. If a desired condition of data checking has been achieved, the *AIM* system then is able to execute the database linking. The linkage of the database is followed according to the ordering of program statements. The *AIM* system decides where to search the required database, when to apply the adaptive value, and how to link the database to complete a single task on each statement. The above described processes are referred to as the database management of the *Adept* robot system.

3.2 Visionware System

A vision system is essential for alignment and inspection tasks in today's assembly equipment [26]. The *Adept Visionware* (Vision Software) system involves usage of two modules to achieve this goal. One module is called the *vision module* and the other is called the *robot module*. The vision module has the ability to measure, inspect, analyze, and recognize the object's image [27-28]. The robot module is used to perform the task of robot motion control. When the exact location of a part has been decided by the vision module, the robot module provides a transformation in between robot location and image

positions. It is this transformation which makes the robot system move a specific distance and rotation angle to achieve an accurate object location.

There are several essential functions that the *Adept* 'window-based' Visionware can provide for vision applications. These various functions are governed by the *'AGS-GV'* system which is the core software that the Visionware uses to complete image analyses. This *AGS-GV* system is shown in Appendix A; each accessory used to support the *AGS-GV* system is also described. These essential functions are boundary analysis, modeling, and vision transformation. The following sections describe the concepts of the Vision-ware's essential functions and the characteristic operations that they can perform.

Boundary Analysis -- Boundary analysis locates objects by looking for bounded areas (a closed string of lines and/or arcs) in the field-of-view, and returns information about those object sizes, locations, perimeters, etc. The boundary analysis decides the boundary of an object by comparing the light intensities between pixel arrays. There are two scales that the *AGS-GV* system can process: the binary scale vision and the gray scale vision [29]. The *binary scale vision* is the simplest method to present an object's image, because the image within the field-of-view is separated into black and white blocks. The determination of the black or white blocks is achieved by comparing the light intensity of a pixel with a certain threshold value. All pixels with values higher than the predetermined threshold value are seen as white, while those below that value are seen as black. Since there are only black and white blocks, the boundaries of an object provide a large intensity drop. This large intensity drop is easily detected by the vision system; so the boundaries of an object can be

correctly decided. Unlike the binary scale vision which uses only black and white blocks, the *gray scale vision* uses *128* levels (from intensity 0 to intensity 127) to represent the image. Each level has a specific light intensity value; consequently, the value that is placed in the pixel will be represented by a particular shade of gray. Commonly, the gray scale vision is not used on the boundary analysis, because its gray level eliminates the large light intensity drop at the object's boundary: which can reduce the accuracy of boundary analysis. A simple demonstration of an object represented by the binary scale vision and the gray scale vision, using a *22×16* pixel array, is shown in *Figure 7.*

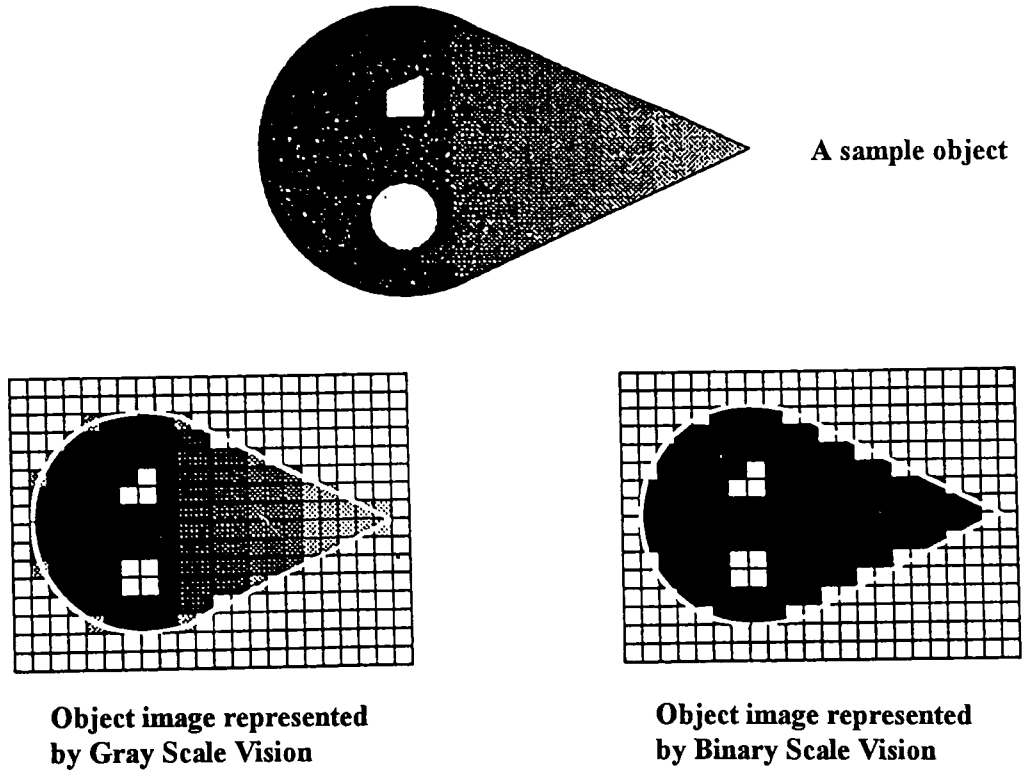


Figure 7. Gray Scale Vision and Binary Scale Vision

Image Recognition -- Image recognition involves two important parts: image modeling and image matching. *Image modeling* allows the vision system to store models of objects; the main purpose of this operation is to compare stored models with real objects in order to recognize those objects [30]. Before an image model can be established, image training must be performed. *Image training* generates an idealized 'vision model' of an object. This idealized vision model portrays individual characteristics concerning the object's size, area, perimeter, or boundary, etc. After sufficient characteristics have been established, this idealized image model is stored in the database as the basic information for image matching. When an object enters the field-of-view, the *image matching* first places the trained vision model in its memory and instructs the vision system to see if this object matches the model correctly. This procedure is performed by comparing the location of the boundary or the image area of the object with that of the stored vision models. Once the image matching has been successfully performed, the robot system is able to correctly recognize this object; then the object's location and orientation are calculated and stored in other databases for the further operations.

Vision Transformation -- Guided vision is, essentially, the process of putting together several pieces of information to create a vision transformation. Vision transformation is used to calculate the relationships in between the vision system and the robot motion system. Since information about an object can be acquired only by the vision system, it is required to introduce the information into the system such that the robot can precisely achieve object location. When a part is presented under the camera, the camera calibration

is first applied to establish the location relationships in between robot base and camera position. The vision system will then provide various vision tools to specify the exact location and orientation of the part. The location databases are stored in the coordinate system of Visionware as the vision location, while the orientation databases are used to calculate the rotating angle of the end effector. Combining the camera calibration, vision location, and orientation information, the controller then can decide the distance and angle between the part and robot base. Consequently, the part can be precisely grasped by the robot system.

There are two general types of vision transformation that the vision system can perform, based on the different camera mounted positions: the stationary-mounted camera and the arm-mounted camera transformations. In the *stationary-mounted* model, the camera is placed in an immobile position. So the distance between the camera and the robot base are maintained at an unchanging value. The vision transformation in this case is only required to calculate the object vision location and end effector transformation. In the second case, the camera is permanently mounted on one of the robot arms; thus the camera can be moved along with the robot's motions. Therefore, all the possible vision transformations must be involved to acquire the correct location. Although these two transformations are slightly different in terms of method of data gathering, the major purpose of the vision transformation is to precisely acquire the location of a part and then guide the end effector to that location. *Figure 8* shows the typical vision transformation of an arm-mounted camera.

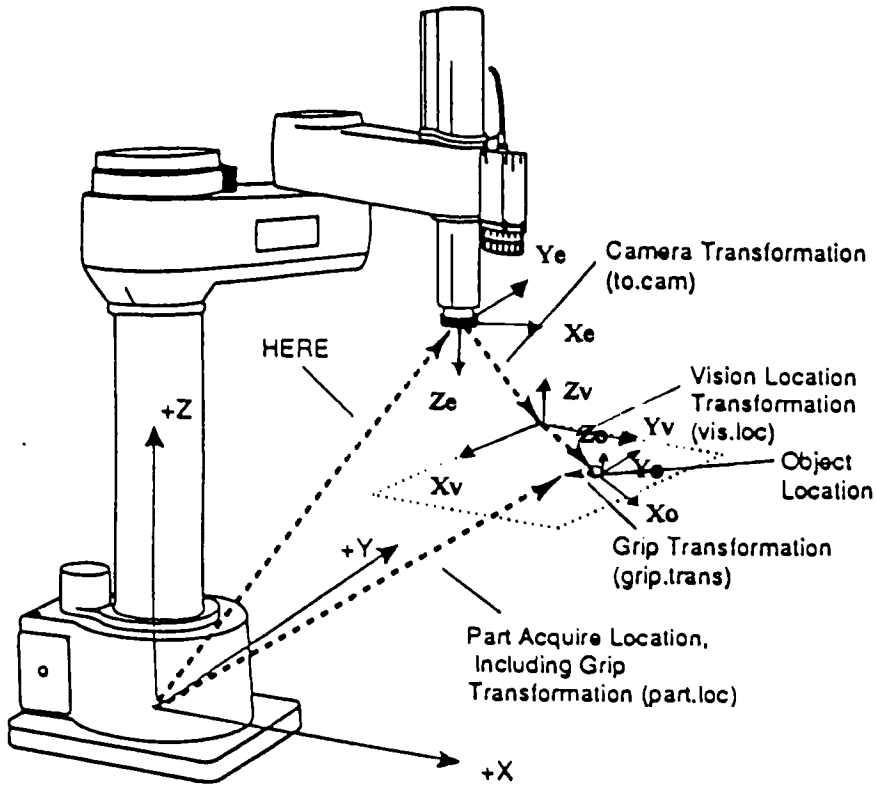


Figure 8. Vision Transformation of an Arm-mounted Camera [28]

3.3 Motionware System

Motionware (Motion Software) comes with a complete set of reference materials that allow the user not only to use, but also to customize at the programming level. Motionware also makes use of vision modules and robot modules in addition to it generating another task, which is position management. The most common action a robot makes at a location is to acquire or place a part. However, these are not the only actions that can be made. The *Adept* Motionware provides ways for controlling most any type of end effectors. In order to acquire or place an object, some basic records are required: to

specify where the part is and how to acquire and place the part. These various records are designated as follows: frame record, location record, and tool record. Each record performs a different task and provides individual information concerning all positions during assembly. The following sections describe the databases that Motionware uses to store the information.

Frame Records -- All robot motion is based on frames of reference and location variables. Location variables uniquely identify a point within a Cartesian space and the orientation of the robot tool at that point. A frame record is used to define the coordinate system which indicates the origin position and all the locations for object placement and insertion. Two types of coordinate system records are used: the world coordinate system and the relative coordinate system. The *world coordinate system* is the primary reference frame of the *Adept* robot system: whose center is at the base of the robot, with *+Z* axis pointing straight up, *+X* axis going front to back, and *+Y* axis going left to right. Unless otherwise specified, the *X, Y,* and *Z* components of a location are actually defined by this primary coordinate system. In many cases, the world coordinate system is insufficient in part acquisition and insertion. For example, all the part locations on an entire assembly may be taught, but, during the actual placement of a part, it is found that exact placement locations have changed; the part feeders may move; the location of assembly may change; or the robot itself may need to be moved or re-calibrated. In any of these cases, retraining the locations in the work-cell will be easier if the majority of location offsets are relative to a secondary coordinate system other than the primary coordinate system. A relative frame is

therefore employed to specify this secondary reference frame. The *relative coordinate system* is a sub-coordinate system, which is defined relative to the world coordinate system. If a relative frame has first been created for the work-cell, and then all part locations had been taught relative to that reference frame: only the points needed to define the reference frame would need to be re-taught. So a relative frame system is more flexible than an absolute world coordinate system.

Location Records -- A location record contains all the required information to indicate the movement of the robot. All the parts' positions within an application unit can be generated and stored in location records. Each location record primarily involves the robot configuration, the digital I/O signal, and the frame record. These sets of data provide the information that the robot must follow when attempting to achieve that location. The robot configuration defines the relationships on robot arms such that the robot enters that location with a particular arm position. The digital I/O signal controls the motion of the end effector and the sensors. The controller simply generates the digital I/O signals to complete a task on that location, such as grasping a part from the feeder, placing a part on the conveyor, or changing the tool for multiple processing, etc. A frame record is used to define the frame system of this location. The frame can be a world coordinate system or a relative coordinate system, depending on the application requirements. The exact location value of the object is associated with the assigned frame record: which makes the motion of the robot become easy to control when achieving the location of a complex work-cell.

Tool Records -- When a robot location is recorded, the actual physical location is centered at the robot tool flange; thus, the motion of the robot is based on the tool flange's position. However, the location that is defined by the center of the robot tool flange is not desirable, because there are always end effectors mounted on the tool flange for different applications. In this case, the actual physical location is not centered at the robot tool flange, but is at the center of the end effector. Once an end effector is attached to the tool flange, the center of the end effector is offset from the tool flange. In order for the recorded locations to be accurate, this offset must be factored out of the location record. A tool record is used in such a case to record the offset information. This offset will be applied so that the tool can correctly operate at a desired location within the work-cell.

3.4 Sequence Editor

Sequence editor is another sub-software under the *AIM* system. It generates a sequence of statements, the usage of which is to control the robot motion, processing flow, or vision operation. Sequence editor can be seen as a kind of traditional programming method, but is a window-based application. Inside, the sequence editor contains a complete collection of statements. Each statement has two basic factors: command and record. Within the sequence editor, *command* is the pre-defined code, which is used to complete a single task. The *record* database must be assigned to its suitable command; for example, motion commands require motion records for guiding the robot; and vision commands need vision records to perform image analysis. Meanwhile, the statement's

order can be organized by the operator, based on the operating procedures. Once the order of the sequence has been defined, the motions of the robot must follow this order to proceed with the assembly flow. The statements used for the sequence editor are listed in *Appendix B* with brief explanations.

CHAPTER IV

END EFFECTOR DESIGN

During assembly processing, the maximum forces acting on the *ICs* are restricted not to exceed *0.05N* (0.011lbf) in order not to squeeze out the adhesive materials used for chip binding. To achieve this goal, the moving distance of the robot in the *Z* direction must be controlled within a tiny range. Although it might be possible to program the robot to move a small increment, while checking the force applied between each movement, the robot vibrations and repeatability could cause the applied force to be unreliable. Therefore, to apply this small force magnitude, an end effector needs to be designed. Since the attachable surfaces of the chips are very small, to grasp or to locate a chip requires a small and effective end effector to perform this task. The end effectors used today are generally of a finger-type mechanism. Using such an end effector would not be desirable, because the grasping surface of the chip is small, especially when surface mounted devices are employed. In addition, the force applied on the chip during the lifting and inserting motions needs to be controlled in order not to break the pin of a chip or squeeze out the solder paste on the printed circuit board. So, it is also necessary to incorporate a force sensor inside the end effector to correctly measure the applied force. Based on the above considerations, the objective of end effector design is to build an attachable robot end effector, to lift and situate the chip on the circuit board without exceeding the acceptable force. Vacuum suction is sought instead of a finger-type end effector for chip assembly. Also, a

magnet-type force sensor will be used to apply and detect the correct force acting on the chip. The following sections describe the vacuum force generation, and the force sensor-based end effector.

4.1 Vacuum Force Generation

The basic theory of vacuum suction focuses on the pressure variance. This pressure variance causes the force difference on an open end such that a net force is obtained. This net force can be utilized to lift plastic or flat surface materials–even large sheets of metal. Commonly, there is a vacuum cup attached on the open end and connected to a vacuum pump, so the pressures inside and outside the cup can be maintained. A special device called *PIAB* is used to generate vacuum suction. The *PIAB* makes use of the *Bernoulli* theorem [31] for high speed air flow in order to produce local pressure variance, so the pressure difference can be obtained. *Figure 9* shows the venturi inside the *PIAB* device.

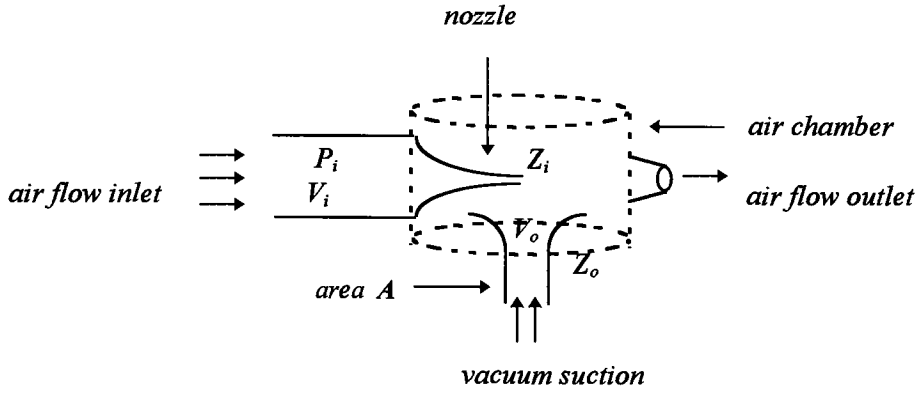


Figure 9. Venturi PIAB

The *Bernoulli* equation is:

$$\frac{P_i}{\rho} + \frac{V_i^2}{2} + gZ_i = \frac{P_o}{\rho} + \frac{V_o^2}{2} + gZ_o \qquad \text{---------- (1)}$$

where

P_i : *air pressure inside the nozzle*
V_i : *air flow velocity inside the nozzle*
Z_i : *altitude at nozzle end*
g : *gravity*
P_o : *atmospheric pressure*
V_o : *outside air flow velocity*
Z_o : *outside altitude*
ρ : *air density*

Since the structure of venturi is very small, Z_i and Z_o are assumed to be at the same altitude, and the air density is assumed to be a constant during operations. Thus equation (*1*) can be re-written as:

$$\frac{\rho \cdot (V_i^2 - V_o^2)}{2} = P_o - P_i \qquad \text{---------- (2)}$$

and the vacuum force F becomes:

$$F = (P_o - P_i) \cdot A = \frac{\rho \cdot (V_i^2 - V_o^2)}{2} \cdot A \qquad \text{---------- (3)}$$

where A is the area of the vacuum cup. From equation (*3*), the vacuum force can be controlled by air inlet velocity V_i. This is performed by an adjustable valve connected between the air pressure source and the *PIAB*. One vacuum cup is sufficient for chip assembly, because the attachable surface of the chip is very small, the chip weight is small, and the robot is only required to lift or insert one chip in each cycle. In order to lift the chip with maximum weight *0.058N* (0.013lbf) and minimum surface width $6\times10^{-3}m$ (0.236in), the diameter of the vacuum cup is designed to be $5\times10^{-3}m$ (0.196in). A safety factor of *2* is

applied to ensure that the vacuum force can correctly lift the chip. Under the standard atmosphere pressure, the valve is adjusted to provide a lifting force *0.116N* (0.026lbf) on the vacuum cup. This force is sufficient for chip lifting and inserting during assembly since the chip weight is only *0.058N*.

4.2 Force Sensor

The basic concept of a *magnet-type force sensor* is to make use of magnetic repulsive forces to provide a desired net force. The magnetic force has a non-linear characteristic, relationship between load and displacement, which can be determined. The structure of this force sensor is composed of two major parts: a force sensing circuit and a force applying device.

The *force sensing circuit* is used to determine the force applied on the integrated circuit components during *PCB* assembly [32]. This main circuit is operated by a *9volt DC* power supply, and can convert the applied force to a signal output determined during calibration. A detail discussion of this force sensing circuit is presented in *Appendix C*. From *Appendix C*, a variable resistor R is the main resistor used to detect the force acting on the chip. The voltage crossing resistor R is compared with the voltage crossing resistor R_2, which is defined as a threshold voltage. By measuring output voltage variances of resistor R, the forces applied on the chip can be determined. Two $50k\Omega$ linear potentiometers are series connected by a metallic slider to represent this variable resistor. The reason for using two linear potentiometers as the variable resistor is to amplify the

resistance difference when measuring the output voltage. *Figure 10* diagrams the positions of these two potentiometers and the structure of for sensor.

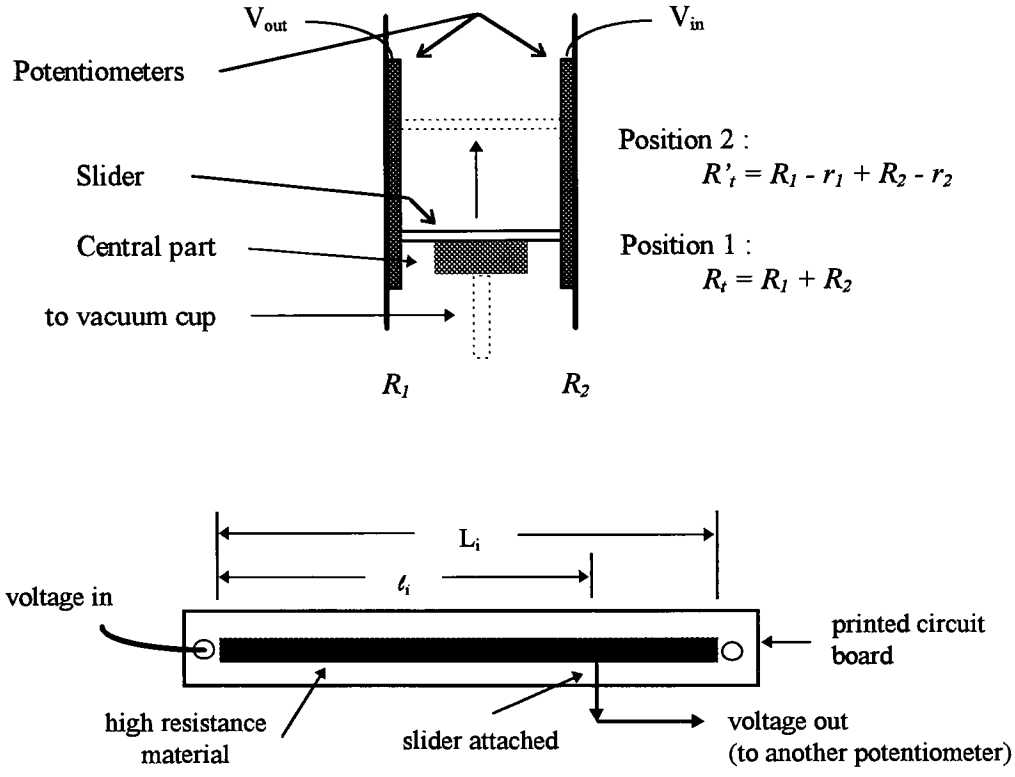


Figure 10. Location of Two Linear Potentiometers and Their Structures

The resistance distributions of two potentiometers are assumed uniform, so the resistance R_i on each linear potentiometer is:

$$R_i = \frac{\ell_i}{L_i} R_{ini} \quad (i = 1 \text{ and } 2) \qquad \text{---------- (4)}$$

where L_i is the total length of high resistance material, ℓ_i is the length from one end of a linear potentiometer to the slider attached position, R_{ini} is the total resistance of a linear

potentiometer, and the subscript i indicates the potentiometer 1 or 2. In addition, the relation between voltage and current is assumed linear. Therefore, the voltage difference between voltage input V_{in} and voltage output V_{out} can be presented as follows:

$$V_{in} - V_{out} = i \cdot R_t \qquad \text{---------- (5)}$$

where R_t is the summed resistance of two linear potentiometers. The current i passing through linear potentiometers is constant; so the voltage output is proportional to the resistance change R_t. When there is no movement of the force-applying device in the Z direction, the slider stays at the initial position, say position 1. The resistance of each potentiometer at this position is R_1 and R_2. The total initial resistance is the summation of the two potentiometers which is equal R_1+R_2, and the output voltage is $V_i = i\ (R_1 + R_2)$. Once the robot moves, a small increment is added to the slider. The slider will then move to position 2. At this position, the resistance of R_1 changes to $R_1 - r_1$ and the resistance of R_2 becomes $R_2 - r_2$, where r_1 and r_2 present the resistance change on R_1 and R_2, respectively. The total resistance R'_t due to this displacement is equal to $(R_1 - r_1)+ (R_2 - r_2)$, and the corresponding voltage at this position is $V_f = i\ [(R_1 - r_1) + (R_2 - r_2)]$. Comparing these two positions, the total resistance change is $-(r_1 + r_2)$ and the voltage difference is $V_f - V_i = -i\ (r_1 + r_2)$. Therefore, the resistance variance is scaled by $r_1 + r_2$, and the quantity of the voltage difference is amplified by this scaled factor. This voltage difference is much easier detected than the voltage changed with a single variable resistor. Therefore, a pair of potentiometers can provide better resolution in measuring the force applied on the chip.

4.3 Force-Applying Device

The *force-applying device* includes two parts: an outside cell and a central part (*Figure 11*). The outside cell is a cylindrical plastic shell used to integrate the whole device. A tool changer kit is mounted on the top lid for multiple application usage. The central part is designed to move independently with respect to the outside part, in the *Z* direction. This independent motion is utilized to generate the motion of the slider, so the applied forces can be determined from the force sensing circuit. The vacuum duct, vacuum cup, and the magnets are also placed on the central part. To achieve the required pressure, the number of the magnets, their placement locations, and their spacing should be considered. There are two pairs of magnets placed inside the end effector. The first pair, located between the bottom of the central part and the bottom lid, with *N* poles facing, is utilized to balance the weight of the central part. The second pair, mounted in between the top of central part and the top lid with *S* poles facing, is a force-applying device. The magnets are all permanent magnets (*PM*), which can provide a stable magnetic flux under normal conditions. Since the weight of the central part is much larger than the force that is required to be applied on the chip, the magnetic strength of pair 1 (*NN*) is larger than that of pair 2 (*SS*). Therefore, the weight of the central part can be balanced by magnetic force, and the force applied on the chip can also be controlled by a small increment. *Figure 11* shows the structure of this force-applying device and the locations of each magnet pair.

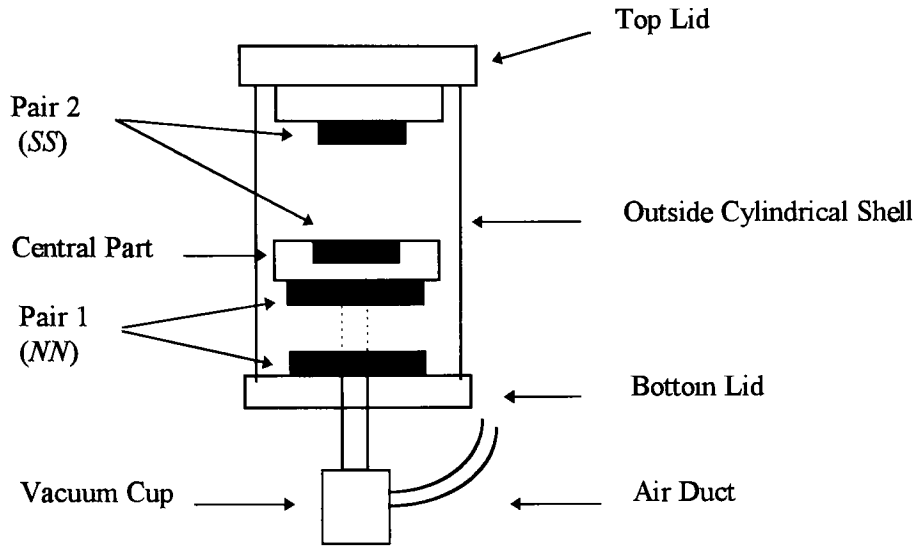


Figure 11. Structure of Force-Applying Device and Magnet Locations

Once the size of magnets and their locations has been decided, the remaining consideration is the distance between each pair. When the robot arm moves downward and the vacuum cup touches the chip, the distance between the *NN* pair increases while the distance between the *SS* pair decreases. The repulsive force of the *SS* pair is increased and that of the *NN* pair is decreased. Therefore, a net downward force is generated on the central part (or chips).

As a means toward optimizing the spacing of the magnet pairs, to ensure that the force can be applied on the chip when the robot arm is moved, the magnetic force which is applied to each pair has to be considered. The magnetic force generally is proportional to $P_1P_2/\mu r^2$, where P_1 and P_2 are the strength of the magnetic poles, r is the distance between two poles, and μ is the property of the medium [33] called the permeability. The magnitude of permeability μ is defined as the permeability of a vacuum. For air, in which most mag-

netic experiments are made, μ is *1.0000004*; and it is approximately unity for most materials [34]. Hence, in the majority of cases, μ may be omitted from the formula. The complete form of the magnetic force can be written as follows:

$$\mathbf{F} \propto \frac{\mathbf{P_1 P_2}}{\mu \mathbf{r}^2} \qquad \text{---------- (6)}$$

or

$$\mathbf{F} = \mathbf{C''} \frac{\mathbf{P_1 P_2}}{\mu \mathbf{r}^2} \quad ; \quad \text{for any unit} \qquad \text{---------- (7)}$$

where C'' is a constant, depending upon the choice of units. The magnetic forces act along the line joining the two poles. Because it is not possible to isolate point poles, equation (*7*) is an ideal law. When applied to actual magnets, it gives results that are only approximately correct [34]. But it is most useful as a basis for the definitions and theory that follow. If proper units are chosen, the constant C'' can be eliminated. For example:

$$1 \text{ dyne} = C'' \frac{(\text{ 1 unit pole })(\text{ 1 unit pole })}{(\text{ 1 for vacuum })(\text{ 1cm})^2} \qquad \text{---------- (8)}$$

whence

$$C'' = 1 \qquad \text{for these units}$$

From the above discussions, a total magnetic force [33] can be formulated as:

$$\mathbf{F_M} = \mathbf{F_{attractive}} + \mathbf{F_{repulsive}}$$

$$= \sum_{i=1}^{12} \frac{\mathbf{P_{NN} P_{SS}}}{\mathbf{r_1}^2} - \frac{\mathbf{P_{NN} P_{NN}}}{\mathbf{r_1}^2} - \frac{\mathbf{P_{SS} P_{SS}}}{\mathbf{r_2}^2} - \sum_{j=1}^{5} \frac{\mathbf{P_{NS} P_{SS}}}{\mathbf{r_j}^2} - \sum_{k=1}^{5} \frac{\mathbf{P_{SN} P_{NN}}}{\mathbf{r_k}^2} \qquad \text{---------- (9)}$$

where

P_{NN} *: one N pole strength of NN pair* $\qquad$ P_{SS} *: one S pole strength of SS pair*

P_{NS} : *one S pole strength of NN pair* P_{SN} : *one N pole strength of SS pair*
r_1 : *distance between NN pair* r_2 : *distance between SS pair*
r_i : *distance between one N pole of NN pair and one S pole of SS pair*
r_j : *distance between one S pole of NN pair and one S pole of SS pair*
r_k : *distance between one N pole of SS pair and one N pole of NN pair*

Based on the strength of P_{NS} equal to P_{NN} and P_{SN} equal to P_{SS}, the first summation term has properly expressed the total attractive forces. The same situation applies to the fourth and fifth summation terms of repulsive force expressions. Due to the fact that magnetism is the function of temperature K [35], the distance is the function of time t during operation, and the minor forces are assumed to be omitted, the resultant force is a theoretical approximation. Therefore, the resultant force is measured experimentally instead of done by theoretical calculations. The measured data is listed in *Appendix D*. Returning the experimental result from *Appendix D*, the approximate pole strength of *NN* and *SS* pairs are *0.008* and *0.003 unit poles* [34], respectively. The distance for *SS* that can significantly interfere with each other is *0.043m* (1.69in). Since the required force acting on the chip is *0.05N*, the distance between *SS* pair decreases to *0.03m* (1.18in) to ensure that the motion of the robot can simultaneously apply force on the chip. At this initial modified distance, the central part weighting *0.023kg* (0.05lbm) is kept at a clearance of *0.016m* (0.63in). The second and third terms in equation (*8*) dominate the magnetic forces because r_1^2 and r_2^2 are much smaller than the other ds^2 terms. With the thickness of magnets and the central part given as *0.005m* (0.19in) and *0.011m* (0.43in), respectively, the above description is calculated in *Appendix E* and shown in *Figure 12*.

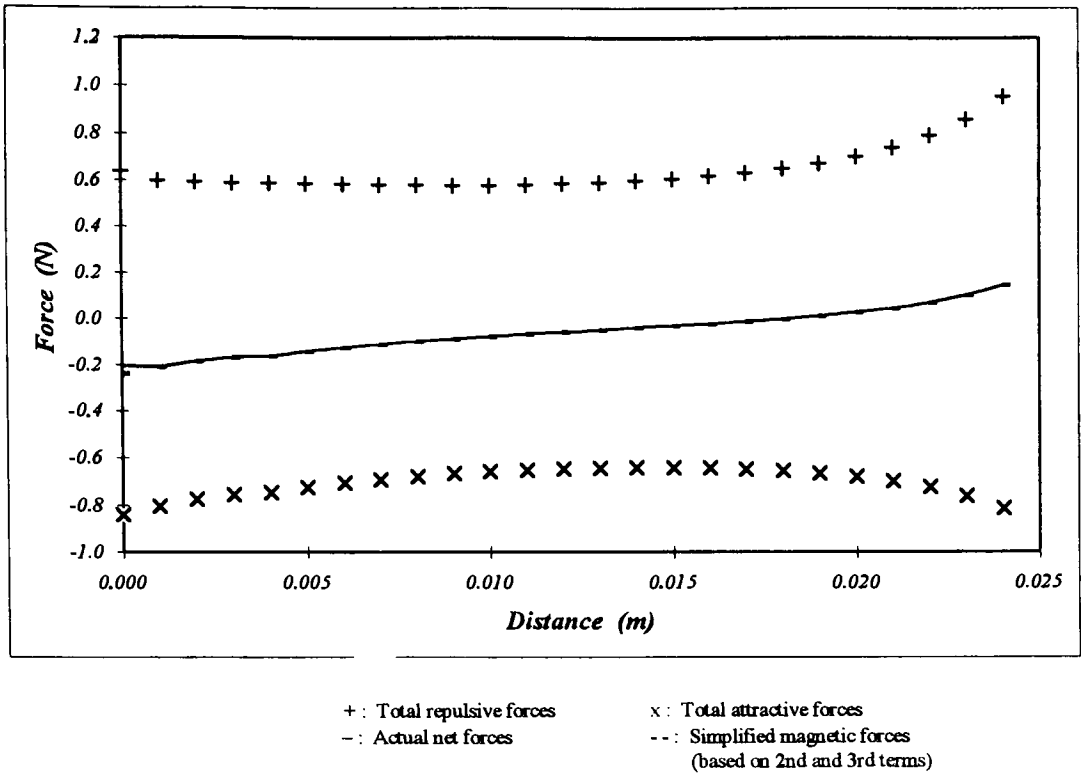


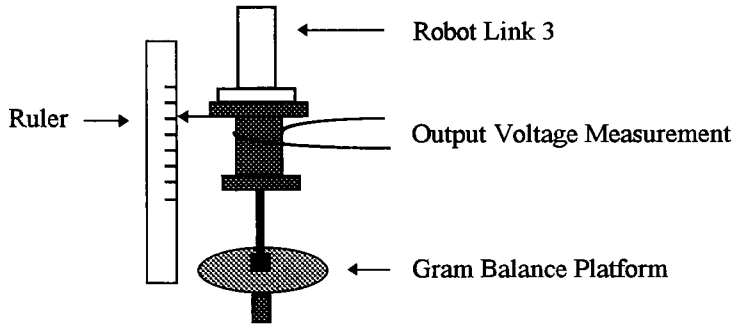
Figure 12. Actual Magnetic Force and Simplified Magnetic Force

An average difference between actual net forces and simplified magnetic forces is *3.06%*, so the above assumption is acceptable. The simplified magnetic forces, therefore, will be used to compare with the experimental results during calibration. The detailed structure of this force-applying device is shown in *Appendix F*.

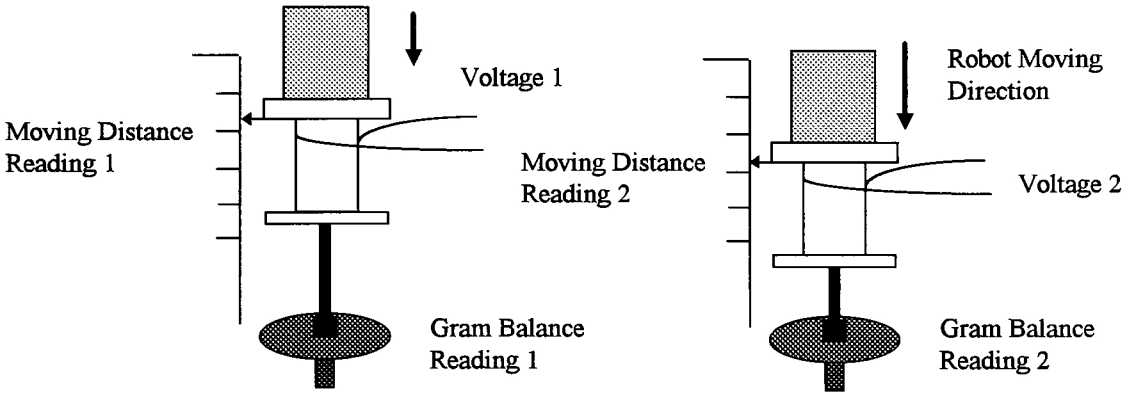
4.4 Force Sensor Calibration

The force sensor calibration initializes the force-applying device before operations and optimizes the end effector system, so the relationships between applied forces and output voltages can be established. A gram balance is incorporated in this calibration to

measure the force acting on the tip of the vacuum cup. During the calibration, the end effector is stationary mounted on the robot arm link 3. For each small increment in the Z direction, the balance of the gram balance is observed. Once the gram balance is balanced, the robot traveling distance, applied force, and output voltage crossing the potentiometers are recorded. *Figure 13* shows the end effector calibration operations.



End Effector Calibration Setup



Operation 1 Operation 2

Figure 13. Calibration Setup and Operation Procedures

The calibration result should have some errors considering that the magnetic force was obtained experimentally. In order to minimize errors and to use the calibration result on a full measurable range, the numerical fitting functions are introduced to interpolate the data. Those fitting functions can be thought of as equations to represent the magnetic force, output voltage, and the relationships of force versus voltage. One of the simplest approaches to data fitting is based on the polynomial least squares [36]. A general polynomial fitting function of degree *n-1* is in the form:

$$y = a_1 + a_2x + a_3x^2 + \cdots\cdots + a_nx^{n-1} \qquad \text{---------- (10)}$$

where $n\text{-}1 \leq m$, the number of data points. Usually, *n-1* is much less than *m*, which means that a low degree polynomial is fitting through a large amount of data. In this case, the coefficients a_k must adjust to minimize the sum of squares of the errors. *MATLAB™* software is used to calculate the coefficients of these polynomial fitting functions [37], and the execution code is listed in *Appendix G*. The recorded calibration values are listed in *Appendix H*. The results and their fitting functions are sketched in *Figures 14*, *15*, and *16*.

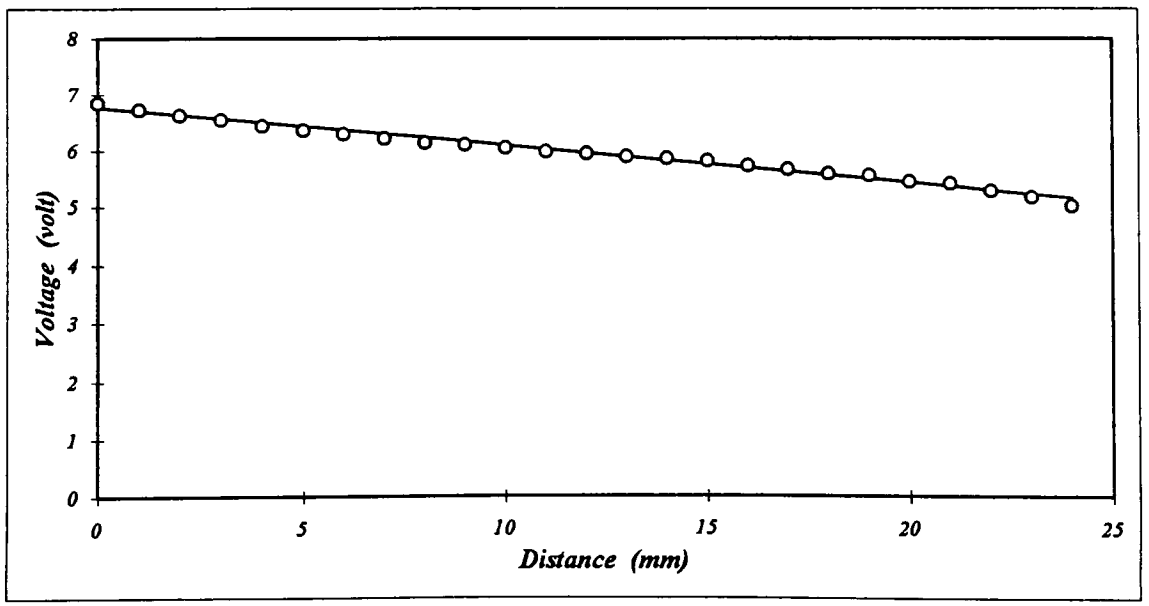


Figure 14. Voltage vs. Distance

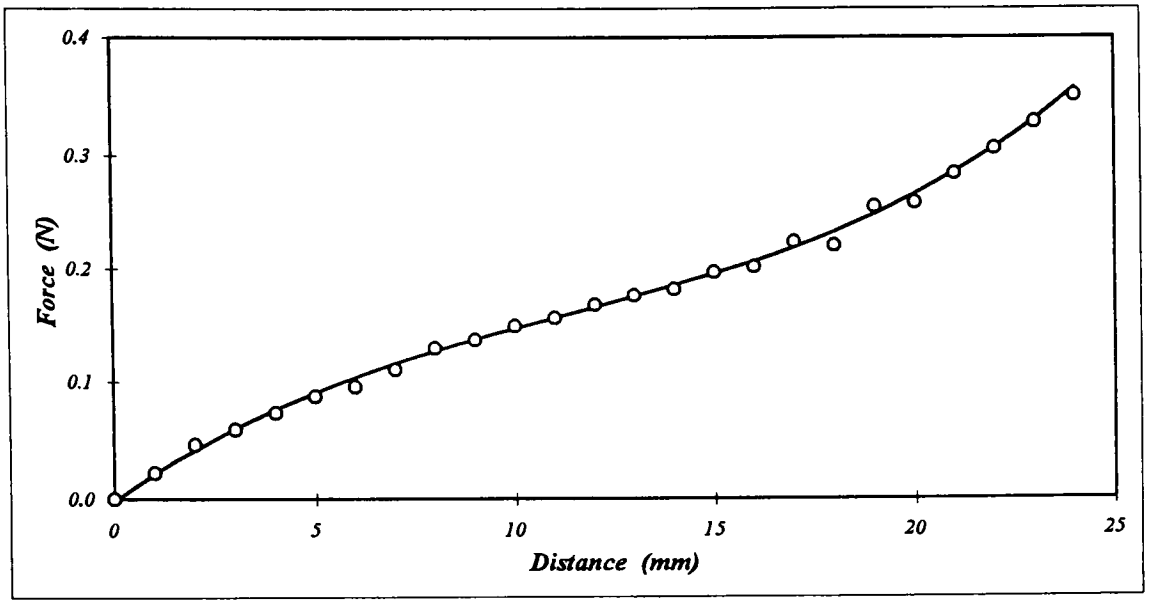


Figure 15. Force vs. Distance

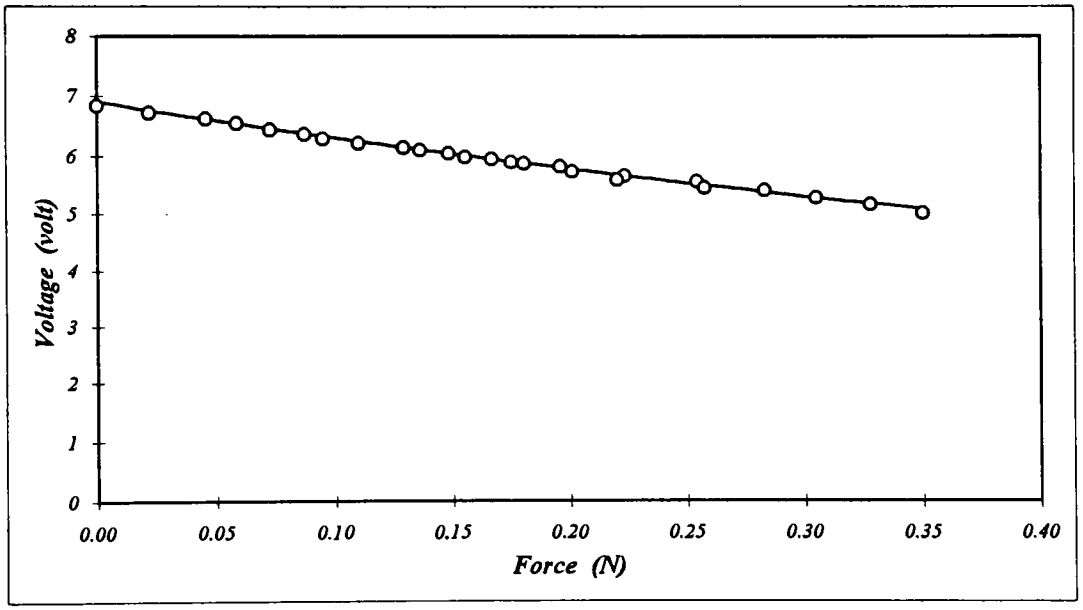


Figure 16. Voltage vs. Force

Figure 14 shows the robot moving distance D and the corresponding output voltage V (VD). Since the central part has the independent motion associated with robot

link 3, one unit robot moving distance provides one unit position changed on the central part. This results in a unit output voltage drop, because the resistance distribution of the linear potentiometer is uniform. Therefore, this *VD* calibration shows a linear characteristic of the robot moving distance and the output voltage. A first degree polynomial fitting function is sufficient to establish the *VD* relationships, and it has the form:

$$\mathbf{V = - 67.5\ D + 6.772} \qquad \text{---------- (11)}$$

with coefficient of correlation r [38]

$$r = -0.99 \qquad \text{---------- (12)}$$

The unit of constant term is *volts*, and the first degree coefficient is *volts/mm*. The coefficient of correlation r indicates that this *VD* figure has good linear relationships between the robot moving distances and the output voltages. This linear equation will be used to predict the robot moving distance according to a voltage value or vice versa. For a *0.025m* (1in) robot moving distance of the central part, the recorded voltage drops from *6.84 volts* to *5.0volts*. The voltage calculated from the fitting function gives *6.772volts* and *5.08volts*, which have tolerances of *0.99%* and *1.6%*, respectively. If an output voltage is located within this range, the robot moving distance can be predicted by this linear equation.

Figure 15 establishes the relationships between the applied force and the robot moving distance (FD). It can be seen that the *FD* correspondence is non-linear, because the magnetic force is proportional to the inverse of the distance square ($F \propto 1/r^2$), and the difference of two magnet pairs. One unit robot moving distance does not result in one unit

magnetic force changed. Since there are two pairs of magnets used in the end effector, the actual force applied on the gram balance is the superposition of these two magnetic forces. Therefore, the *FD* figure shows a non-linear curve. It is necessary to use a higher order fitting function to represent this characteristic. The equation is in the form:

$$\mathbf{F = 3.8644E^{-5}\, D^3 - 1.316E^{-3}\, D^2 + 2.3915E^{-2}\, D - 1.3998E^{-3}} \quad \text{---------- (13)}$$

At the desired applied force of *0.05N*, the robot moving distance calculated from equation (*12*) is *2.577mm* (0.103in). Comparing this with the experimental result of *2.6mm* (0.104 in) the deviation is *0.88%*. This equation is therefore suitable for applied force determination for a known robot moving distance.

Figure 16 displays the objective of force sensor calibration. The output voltage used to measure the accurate applied force on the chip is determined by reading this *VF* figure. Returning the fitting function coefficients from *Appendix G*, the fitting function is:

$$\mathbf{V = 2.8086\, F^2 - 6.185\, F + 6.9017} \quad \text{---------- (14)}$$

For the maximum acceptable force of *0.05N*, the experimental output voltage *6.58volts* was recorded. The polynomial fitting function gives *6.6volts*. Both values are fairly close and the difference is *0.3%*. This fitting function therefore will be incorporated to measure the output voltage fluctuations when chip assembly is proceeding. The above acquired voltage is set as threshold voltage *V*. This threshold voltage will be applied to generate a digital signal on the robot motion control. Once the chip is lifted or inserted by the robot, the output voltage, say *V'*, will be compared with this threshold value. If *V'* is smaller than *V*, then there is no current passing through the *LED* of the force sensing circuit. The robot

arm, in this case, will keep moving to ensure a total contact with the chip. If V' achieves the value V, a current is generated to pass through the *LED* and the force sensing circuit will actuate an output digital signal to stop the robot motion. Meanwhile, the solenoid is opened or closed to provide the required air flow through the device *PIAB* to provide the necessary vacuum suction, so that the chip can be lifted or released at the desired location with a correct amount of force. Therefore, a proper force can be obtained by comparing the output voltage with the threshold voltage.

According to the above descriptions, the objective of designing and building a force sensing robot end effector has been achieved. Each part of the end effector is integrated to perform the force application task; its performance will be described later in *Chapter VI*. The detailed circuit connection of the assembly system is sketched in *Appendix I*.

CHAPTER V

PRECISION DETERMINATION

To determine if an assembly system can meet an acceptable standard of reliability, the accuracy of the system must be predicted before operations commence. This accuracy is referred to as the *precision determination*: which can be defined by some specified values such that total errors can not exceed those values. Since integrated circuit component assembly processing relies strongly on vision guidance, the precision determinations can be divided into two parts: assembly accuracy and vision accuracy. *Assembly accuracy* is the overall goal that the integrated circuit component assembly processing is going to achieve. It also defines the limits of the largest acceptable error during operations. In order to achieve this goal, the structure of the integrated circuit component should be taken into consideration. Since the structures of the *SMD* and *DIL* are totally different, the accuracy determinations are set separately by considering their pin pitches and pin widths.

Vision accuracy provides the fundamental accuracy information that the vision system can achieve when an object is inspected. The vision accuracy is a product of the optimum vision resolution, which is dependent on the lens choice, camera mounting position, and pixel array. To achieve the desired vision accuracy for this small scale integrated circuit component assembly, the proper choice of each component should be taken into consideration. Once the optimum vision resolution has been determined from these fac-

tors, the vision accuracy can be calculated.

5.1 Assembly Accuracy

Assembly accuracy must consider the dimensions of *DIL*, *SMD*, and *PCB* in order to determine the tolerance of the process. Before performing the assembly accuracy assessment, two terms are required to be defined: pin pitch and pin width. The pin pitch and pin width give the basic information in determining the assembly accuracy. The *pin pitch* is the distance measured from the center of one pin to another. It is used to indicate the spacing between two pins. The *pin width* is the width of the pin mounted on the integrated circuit component carrier. The definition of pin pitch and the pin widths is graphically demonstrated in *Figure 17.*

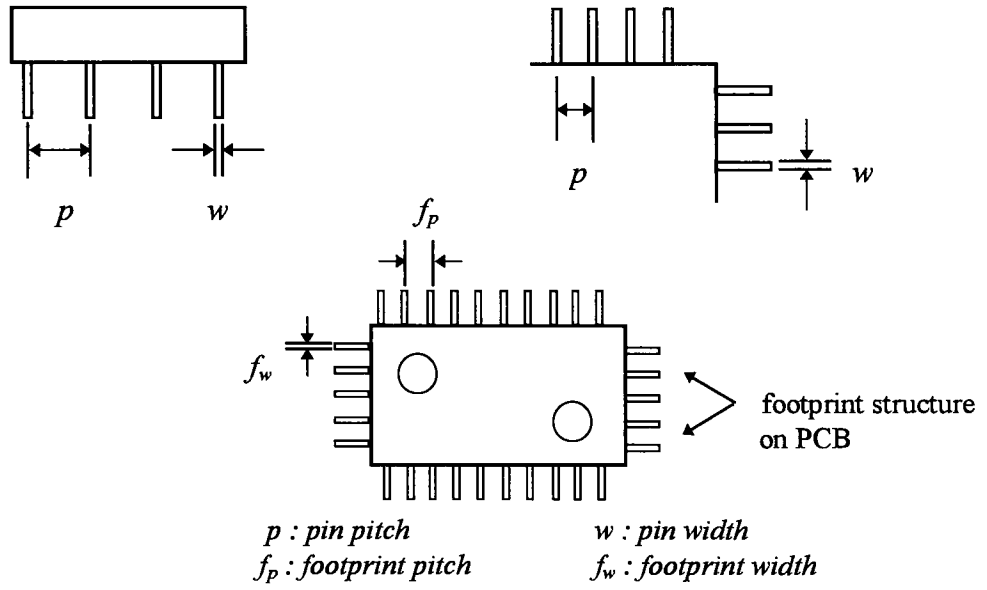


Figure 17. Pin Pitch and Pin Width

For the *DIL*, the pin pitch and pin width are *2.5mm* (0.1in) and *0.5mm* (0.02in), respectively. Also, the aligned holes on the *PCB* used to locate the *DIL* are *1mm* (0.04in) in diameter with *2.5mm* (0.1in) spacing between each hole. The pin tip must be fitted into the holes for successful insertion, so the assembly accuracy of a *DIL* involves the specification of translation and rotation tolerances. The pin pitch and the pin width of a *SMD* are all *0.5mm* (0.02in); the footprint on a *PCB* used to locate a *SMD* has the same dimension, *0.5mm*, in both width and spacing. Since the pin pitch and pin width of a *SMD* are smaller than a *DIL*, the *SMD* is required to meet a higher assembly accuracy than a *DIL*. In *Appendix J*, the assembly accuracy of *DIL* and *SMD* is described in detail. Returning the tolerance values from *Appendix J*, the *DIL* must be located within *±0.3mm* (0.012in) in the *X* direction and *±0.2mm* (0.008in) in the *Y* direction with a maximum $\pm 2^{o}$ rotation tolerance, while the *SMD* has to be placed within *±0.25mm* (0.01in) in translation and $\pm 1.38^{o}$ in rotation tolerances. In addition, a mirror is incorporated to per-form the *DIL* inspection. The mirror reflects the pin images of a *DIL* into the camera, so the vision system can detect the shifting and misalignment errors of the pin. The mirror setup and image transfer calculation is obtained from *Appendix J-3*. Returning the results, the pin alignment accuracy is *±0.16mm* (0.0064in) in the *X* direction, *±0.2mm* (0.008in) in the *Y* direction, and *±0.2mm* in the *Z* direction.

5.2 Vision Accuracy

The camera system used in the integrated circuit component assembly is a medium-resolution *Panasonic GP-CD 40* camera with a 501×485 pixel matrix. A dominant factor that affects the vision accuracy is the *camera resolution*. Generally, there are three sources affecting the camera resolution: image surface pixel number, camera mounting altitude, and lens choice. Since the camera used for integrated circuit component assembly is a *GP-CD 40* camera, the pixel matrix inside the camera remains the same, so the remaining consideration is the proper choice of camera lens and the camera mounting altitude. For the proper *lens choice*, there are only two lenses equipped on the camera: a *50mm* lens and a *25mm* lens. Although the *50mm* lens can provide a larger image size than a *25mm* lens at the same mounting altitude, the minimum focusing distance, *1.3m*, is much larger than the robot's reachable distance in the *Z* direction. The *25mm* lens is therefore applied to determine the resolution. The *camera mounting altitude* has an influence on the field-of-view which encloses all the areas that a camera can inspect. The area of a field-of-view can be adjusted by different mounting altitudes. If the same lens is used, a higher mounting altitude commonly provides a larger field-of-view than does a lower one. But, in this case, an object will appear to be smaller than it does when seen by a camera mounted at a lower mounting position. Therefore, a small scale object, for example, the pin width or spacing of a *SMD*, requires a lower mounting altitude. In order to inspect such small scale objects as accurately as possible, the camera is mounted at the bottom end of the *Adept* robot link *2*. The altitude at this mounting position is *0.35m* (14in) from the working plane, and a

62×58 mm² (5.75in²) area of the field-of-view is acquired. At this camera mounting altitude, and with the *25mm* lens, the camera resolution of *0.1237mm/pixel* (0.005in/pixel) is obtained. The resolution calculation is returned from *Appendix K*. This resolution gives the vision system *±0.1237mm* (0.005in) and *±0.122°* in translation and rotation accuracy, respectively. Comparing the vision accuracy and assembly accuracy, the vision system has sufficient precision to meet the integrated circuit component assembly requirements. In *Figure 18*, the different lens and various mounting altitude influences on the resolution are shown.

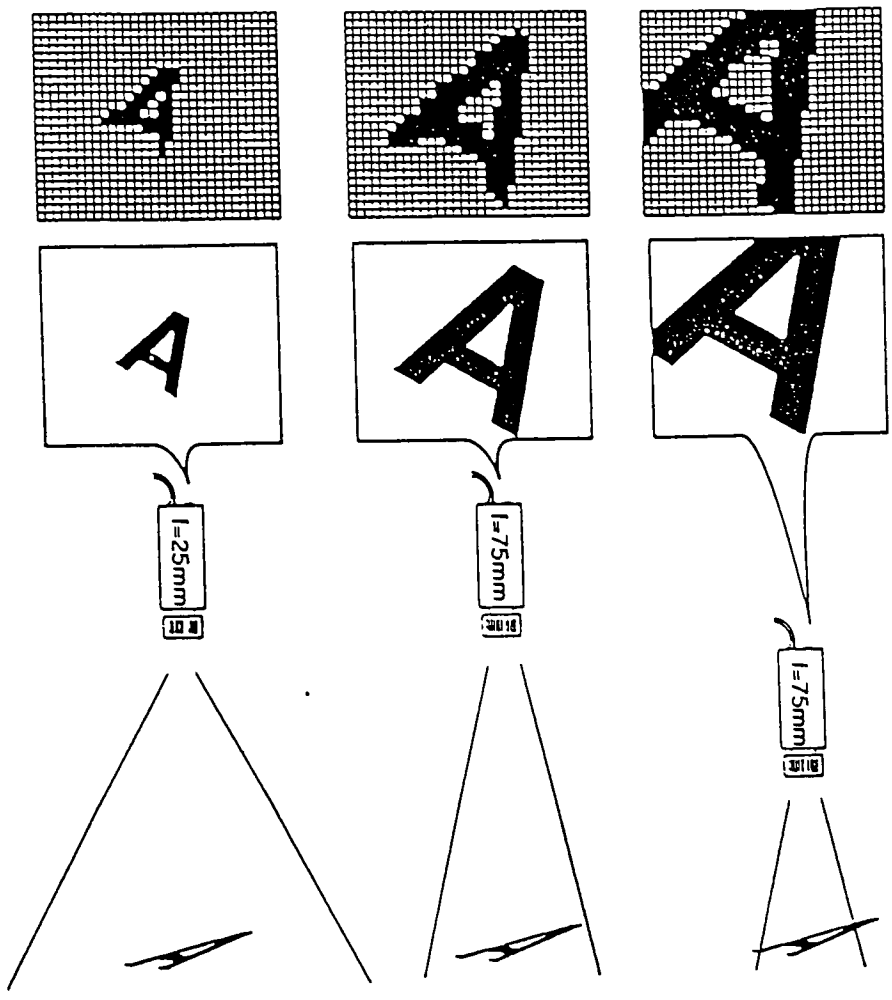


Figure 18. Camera Resolutions of Different Altitudes and Lenses [28]

CHAPTER VI

DISCUSSION of SYSTEM PERFORMANCE

Integrated circuit component assembly processing was run for *30* trials by using the complete vision database records and the final modified code; listed in *Appendix L*. Each trial involved three assembly operations: two *DIL*s and one *SMD*. During assembly process, the integrated circuit components are randomly placed on a whit table, and the *PCB* is transported to a bounded area but not stops at a precise location. Within the trials, the operating cycle time of each step was recorded. The average lift-and-insert cycle time of a *DIL* and a *SMD* were *24* seconds and *27* seconds, respectively. The *SMD* operating cycle time is longer than a *DIL*'s cycle time because the *SMD* has a larger number of pins than a *DIL* has; this requires a greater amount of vision operations. For a complete assembly cycle, the average time is nearly *75* seconds. This time was due to the robot motion, vision operations, and the picture-taking delays. The delay due to the robot motion came from its slow operating speed: *50mm/sec* (2in/sec). Reduction of operating speed was done to prevent abrupt motion causing robot vibration, from affecting the assembly accuracy. Delays during picture-taking were necessary, so that the robot motion could be totally settled before a picture was actually taken. Although time-consuming, this delay tends to enhance the accuracy of the vision operation. A large amount of time was spent on the vision operations, because a large number of databases and the complicated vision software communication were required to perform assembly tasks. This time might be reduced if faster software communication was possible. Considering the above delays, the

operating cycle time generally could be improved by increasing the robot speed, and decreasing the time delay caused by the picture-taking process. The operating time of integrated circuit component assembly is shown in *Table 3*.

Table 3. Assembly Cycle Time

Recorded Time	DIL	SMD
Vision Operations†	*12s (84%)*	*16s (59%)*
Picture-Taking Delay	*0.5s × 4*	*0.5s × 4*
Robot Motion	*10s (16%)*	*9s (41%)*
Sum	*24s × 2*	*27s × 1*
Total Cycle Time	*75s*	

During the integrated circuit component assembly processing, it was observed that there were two sources that tend to affect overall accuracy: vision operation accuracy and robot operation accuracy. Since the desired vision accuracy had been previously calculated, the practical vision operation accuracy was influenced by some environmental factors, such as: robot vibration, robot repeatability, lighting strength, lighting refection direction. Therefore, the vision operation accuracy recorded whether the vision system was successfully performing the required tasks. The robot operation accuracy recorded the guidance accuracy and the end effector accuracy of the system.

The vision system could recognize the position of a integrated circuit component with an error of ± 1 pixel. In a $3596mm^2$ field-of-view and a 501×485 pixel array, this error corresponds to $\pm 0.12mm$ (0.0048in) in translation, and $\pm 0.12^o$ rotation tolerances.

† : Vision operations involve chip recognition and inspection. The values listed are the total vision operation time.

These values represent the position accuracy in recognition of any feature of the integrated circuit component. *Vision operation accuracy* involved integrated circuit component recognition and pin inspection. For the *DIL* recognition, there were two *DILs* that could not be recognized; and additionally, one *DIL* was recognized, but had the wrong rotation direction. The images of these three integrated circuit components, when acquired by the vision system, were slightly different from the model images. The cause for such errors can be explained as being due to the robot repeatability and the lighting direction. The robot repeatability caused the actual assembly location to be shifted from the desired position when the picture was taken. As the lighting source was mounted on the robot, this position shifting generated unnecessary shadows. The vision system took those incorrect shadows to be a part of the integrated circuit component, so the image mismatching occurred. This image mismatching also happened in the case of two *SMD*s, which caused wrong rotation direction detecting. Considering that the *SMD*s, with their flat structure, were randomly placed on the working table, the indication of correct rotation can only rely on the recognition of one corner. A slight shadow at that corner would affect the vision recognition, because the vision system was sensitive to lighting strength. This unnecessary shadow was also related to the kind of robot position shifting described in regard to the *DIL*. However, an overall average *94%* of integrated circuit component recognition accuracy was obtained for both *DIL*s and *SMD*s.

Meanwhile, the pin inspection of the *DIL* was performed by using a mirror, and a back lighting table was used for *SMD* pin inspection. Although the mirror reflects only *10-30%* of the lighting source [39], this amount of light is sufficient for the vision system to

operate. The vision system correctly counted the pin number when the *DIL* was inspected. For the pin pitch measurement, the best vision operation accuracy was *0.13mm/pixel.* This result was a slight deviation from the ideal, and caused the two error cases. For the *SMD*, one error in terms of pin counting and four errors due to pin pitch measurement occurred. These were caused when the center location of the *SMD*'s image shifted. Once the center location shifted, the pin tip found by the vision system was not at the correct position. A vision tool overlap occurred, so the pin number was counted incorrectly, and the pin pitch exceeded a pre-set standard range. The above errors can be avoided by increasing camera resolution, which can be done either by using an image surface with higher pixel number or by reducing camera mounting altitude. *Table 4* shows the operating accuracy of the vision system.

Table 4. Operation Accuracy of the Vision System

Vision Accuracy	DIL		SMD	
	Translation	*Rotation*	*Translation*	*Rotation*
Desired Accuracy	*0.12mm*†	*0.12°*	*0.12mm*	*0.12°*
Operating Accuracy††	*0.13mm*	*0.13°*	*0.13mm*	*0.13°*
Test Number	*60*		*30*	
Chip Recognition	*95 %*		*93 %*	
Pin Inspection	*96 %*		*83 %*	
Average Accuracy	*95.5 %*		*88 %*	

Robot operation accuracy was the second factor affecting on the integrated circuit component assembly accuracy. It involves two factors: robot accuracy and end effector alignment accuracy. Once the location has been determined by the vision system, the lift-

† : This is an average value returning from Appendix K: Camera Resolution.

†† : Operating accuracy is based on multiple measurements of chips, which are stayed at the same locations.

ing and inserting motions are performed by the robot system. There commonly exists repeatability tolerance of the robot system; so a position shift tends to occur even though a high precision vision system is used. The observed accuracy of the *Adept* robot system, in both *X* and *Y* axis, was *0.19mm*† (0.0076in).

The end effector alignment accuracy is another factor affecting integrated circuit component assembly accuracy. After the end effector had been set up, the center of the end effector did not coincide with the robot's *Z*-axis. This misalignment was caused by the end effector producing process. It also generated vacuum cup shifts when the robot's link 3 rotated. The smallest misaligned distance was *0.76mm* (0.03in) in *X* direction and *1.19mm* (0.048in) in *Y* direction, within a rotation angle of *0* to *90* degrees. This inaccuracy of the end effector was adjusted by defining an offset for the robot system and operating the end effector at *0* degree during assembly. This offset provided a good re-calibration†† of the end effector misalignment, and moreover eliminated some uncertainties when high precision was required. The robot system therefore acquired an overall *93%* of accuracy on integrated circuit component lifting and inserting operations. For the *7%* unsuccessful operation, an average shifted distance of *0.3mm* (0.012in) and rotation angle of *0.5°*††† was recorded. This shifted distance could be improved by employing a robot system with better accuracy, and the proper end effector manufacturing. *Table 5* synthesizes the above discussions. It should be noted that the values given here were only validated when the robot system was completely settled. *Figure 19* shows the measured values of integrated circuit component assembly performance.

---

† : This mean value is observed from Motionware location databases.

†† : An offset defined in Motionware tool record.

††† : Mean value recorded from Motionware location database. Trn: 0.3 ± 0.1mm. Rot: 0.5 ± 0.07 deg.

Table 5. System Performance Accuracy

Accuracy	X (*mm*)				Y (*mm*)			
Vision (operating)	*0.13*				*0.13*			
Robot	*0.19*				*0.19*			
End Effector								
Angle (deg.)	*0*	*90*	*180*	*270*	*0*	*90*	*180*	*270*
Tolerances	*0.7*	*0.76*	*0.85*	*0.76*	*1.1*	*1.19*	*1.3*	*1.19*
Avg. End Effector	*0.73 (0-90 deg.)*				*1.15 (0-90 deg.)*			
Re-cal. End Effector	*0.21mm*				*0.33mm*			
Average Tolerance	*0.3mm*				*0.5°*			

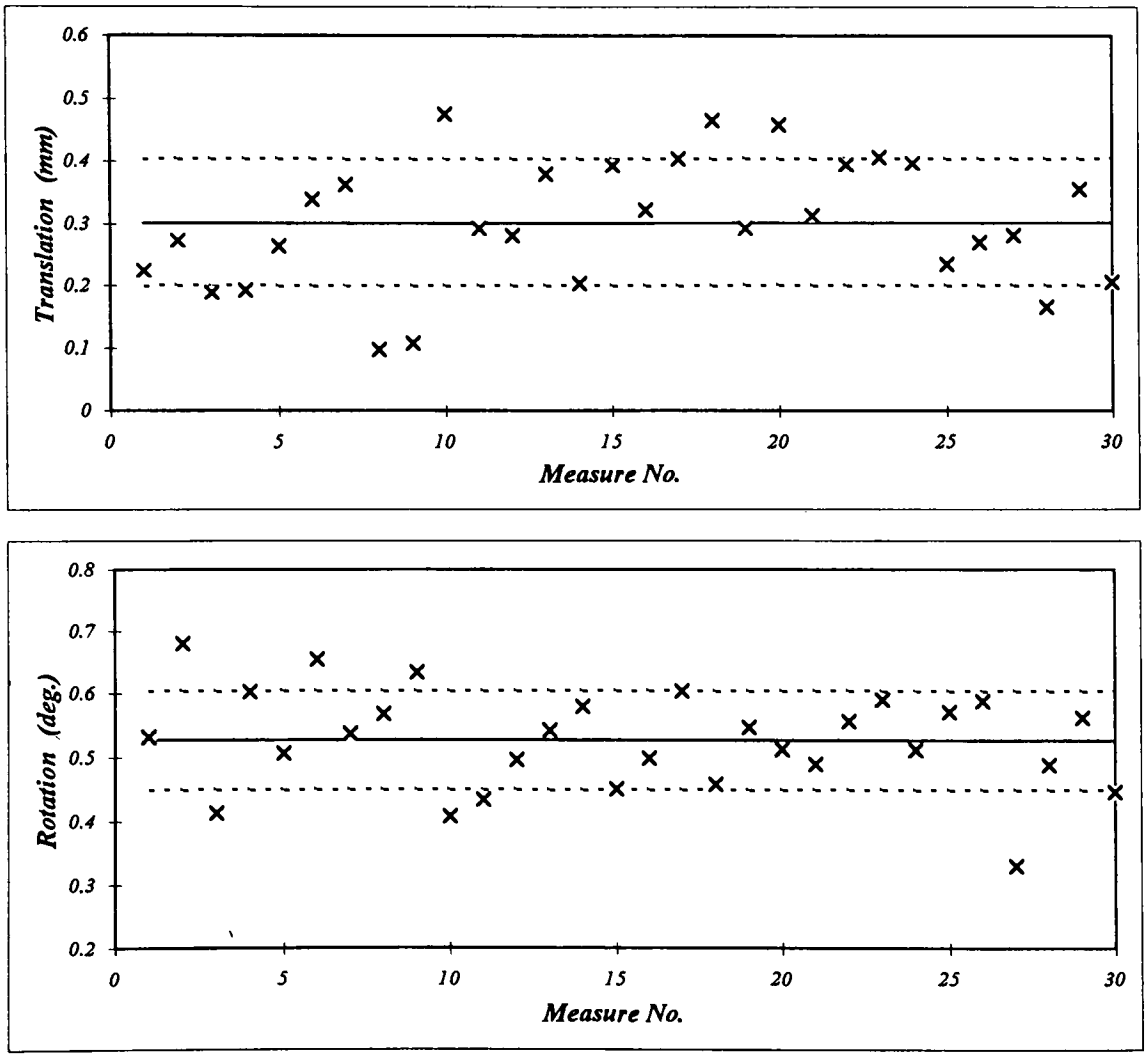
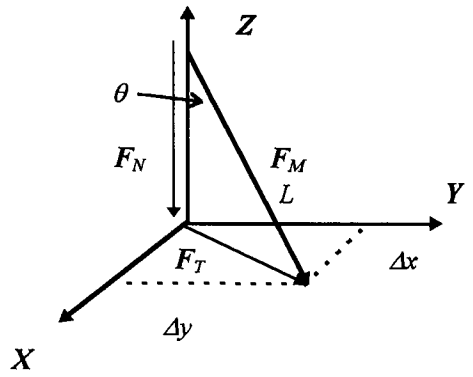


Figure 19. Integrated Circuit Component Assembly Performance

In addition to the vision operation accuracy and robot accuracy, performance of the force sensing device must also be discussed. Digital signals were properly generated by the force sensing circuit to actuate the *PIAB* device for those measurements that were correct, and the integrated circuit component was lifted from and inserted to its correct location smoothly. The voltage output range for correct force measurement was from *6.6 volts* to *6.57volts*, which corresponded with the force *0.0499N* (0.011lbf) to *0.055N* (0.012lbf) calculated from the polynomial fitting function. These forces deviated an average of *5.1%* from the desired force *0.05N*. This error percentage indicated that the polynomial curve fitting method rendered a good approximation in data interpolation, and the results were acceptable. For some cases of incorrect measurement, the force exceeded the desired operating force. This was due to end effector misalignment. With misalignment *0.76mm* and *1.19mm* in the *X-Y* plane, and the length of pointer being *95mm* (3.8in), this produced an angle of 0.84^{o} along the *Z* plane and the vacuum cup. The actual force applied on the integrated circuit component was the *Z* component of the magnetic force, so a larger force was required than the correct one. *Figure 20*, on the next page, shows this *Z* component force, based on an un-aligned end effector pointer.

By converting the *Z* component forces from the output voltages, an average applied force of *0.052N* (0.0116lbf) results, which exceeds the desired force by *4%*. A tangential force of *0.00076N* (0.00017lbf), due to this misalignment, was generated. Though this tangential force was small, it adversely affected on the *SMD* placement accuracy. In addition, the force calibration slope obtained from the force versus voltage



where
F_M : *Magnetic Force*
F_N : *Normal Force*
F_T : *Tangential Force*
Δx : *X error (0.76mm)*
Δy : *Y error (1.19mm)*
L : *Pointer length (95mm)*
θ : *Angle between F_M and F_N due to misalignment*

Figure 20. Applied Force Error Resulting from Misalignment

figure was flat. The voltage difference between the initial and the threshold voltages was only *0.27volts*, which caused some un-certainties on force measurements. The flat slope might be increased by using higher resistance potentiometers, or by series connecting more potentiometers as the force sensing resistor to increase the voltage variances. Though the above two sources resulted in slight output voltage fluctuations, the force sensing circuit, however, did measure the applied forces with an overall accuracy of *90%*.

Summarizing the above discussions, the goal of a vision system assisted robot system for a small scale integrated circuit component assembly was achieved. Parts acquisition and recognition was also successful through simply applying various vision tools, so the part recognition will not be the bottle-neck when objects are randomly introduced to the work cell. It is evidence of the power of the vision system that it is typically possible to reduce location uncertainty. With the aids of the vision system, complex feeder or other costly precision equipment are not required. Improved system cycle times, as well as im-

proved overall accuracy, will be achieved with a higher resolution *CCD* camera, a faster vision and communication system, and a better repeatability of the robot system. In addition, the magnetic-type force sensor does a satisfactory job of measuring critical forces and giving exact feed back to the robot controller. This end effector provides protection for the IC integrated circuit component, and it is hoped that this magnetic-type end effector can be practically used in the industrial work-cell with some improved modifications. In the future, machine vision technology will be the most appropriate solution to many factory automation problems, because it does offer a cost-effective alternative to many repetitive or hazardous jobs. This innovative *MVS* and its multiple purposes, therefore, can lead to industrial success [40].

REFERENCES

1. Pawel Siberring, "Vision System: A Key Technology for Chip Assembly", *Solid State Technology*, September, 1994, pp.79-82.

2. Tim Pryor, "Electro-Optical Inspection and Machine Vision", *Applied Machine Vision Conference*, Ohio: Cleveland, April 7-8, 1982, MS82-182.

3. Ware Myers, "Industry Begins to Use Visual Pattern Recognition", *Computer*, Vol. 13, May, 1980, pp.21-31.

4. Gerald J. Agin, "Computer Vision System for Industrial Inspection and Assembly", *Computer*, Vol. 13, May, 1980, pp.11-20.

5. Gary G. Wagner, "Vision Methodology As It Applies to Industrial Inspection Applications", *Applied Machine Vision Conference*, Ohio: Cleveland, April 7-8, 1982, MS82-183.

6. Jim Field, John Payne, and Chris Cullen, "SMD Placement Usmg Machine Vision", *Electronic Packaging & Production*, January, 1986, p.128.

7. J. L. Davies and K. F. Gill, "Automated Bumper Assembly Using A Vision-Guided Robot", *Proceedings, International Mechanical Engineers*, Vol. 207, 1993, p.61.

8. Thurston Brooks, "Adaptive Robot and Sensory System for Attachment of Electrical Connectors to Solar Arrays", *Applied Machine Vision Conference*, Ohio: Cleveland, April 7-8, 1982, MS82-185.

9. John E. Trombly, "Small Part Sorting and Inspection Using Machine Vision", *Applied Machine Vision Conference*, Ohio: Cleveland, April 7-8, 1982, MS82-180.

10. Arturo A. Rodriguez and O. Robert Mitchell, "A vision System for Manufacturing Applications Under Moderately Unconstrained Conditions", *IIE Transactions*, Vol. 25, No. 4 July 1993, pp.15-25.

11. Bruce W. Hueners and Douglas M. Day, "Advances in Automated Wire and Die Bonding", *Solid State Technology*, March 1983, pp.69-76.

12. J. Lau, J. R. Baker, D. Rice, and B. Shaw, "Solder Joint Reliability of Fine Pitch Surface Mount Technology Assemblies", *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, Vol. 13, September, 1990, pp. 534-543.

13. Michael L. Baird, "Sight-I: A Computer Vision System for Automated IC Chip Manufacture", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 2, February, 1978, pp. 133-139.

14. C. H. Chen and A. C. Kak, "A Robot Vision System for Recognizing 3-D Objects in Low-Order Polynomial Time", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 6, Nov./Dec. 1989, pp. 1535-1563.

15. A. J. Vayda and A. C. Kak, "A Robot Vision System for Recognition of Generic Shaped Objects", *CVGIP: Image Understanding*, Vol. 543, No. 1, 1991, pp.1-46.

16. K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, "Thresholding", *Robotics: Control, Sensing, Vision, and Intelligence*, 1987, p.357.

17. B. Paul and C. J. Maguire, " Integrated Circuit Chip Gripper", Course Project, *Introduction to Robotics*, NY: Rochester Institute of Technology, November, 1992.

18. M. J. Chung and C. S. G. Lee, "An Adaptive Control Strategy for Computer-Based Manipulators", *University of Michigan, Ann Arbor, Tech. Rep*. 10-82, August, 1982.

19. J. Y. S. Luh, "An Anatomy of Industrial Robots And Their Controls", *IEEE Transaction Automation Control*, Vol. AC-28, February, 1983.

20. Lee E. Weiss, "Dynamic Sensor-Based Control of Robots With Visual Feedback", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 5, October, 1987, pp. 404-417.

21. C. Lea, "IC Components", *A Scientific Guide to Surface Mount Technology*, Electrochemical Publications Limited, Scotland: Ayr, 1988, Ch. 2.

22. K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, "Image Acquisition", *Robotics: Control, Sensing, Vision, and Intelligence*, 1987, pp.297-301.

23. J. Critchlow, "Charge-Coupled Devices", *Introduction to Robotics*, NY: Macmillan Publishing Company, 1985, Ch. 9.

24. "Adept Vision Introduction", *Adept Vision Ware User's Course Manual, Adept Technology Inc*., CA: San Jose, Sec. 2.

25. Chuck Raymond, Max Donath, and W. Richard Olson, "Problems of Vision Directed Robots in an Unstructured Parts Handling Environment", *Proceedings, International Symposium on Industrial Robots*, Vol. 2, 1983, 14-1:14-3.

26. Brian M. Atkinson, "Some Applications of On-line Vision Sensing in Industry", *Intelligent Robots: 3rd International Conference on Robot Vision and Sensory Controls*, November, 1983, Pt1, pp. 274-279.

27. G. H. Granlund, H. Knutsson, C-J Westelius, and J. Wiklund, "Issues in Robot Vision", *Image and Vision Computing*, Vol. 12, No. 3, April, 1994, pp. 131-148.

28. "Getting Information From AdeptVision AGS-GV", *AdeptVision AGS-GV User's Guide*, *Adept Technology Inc.*, CA: San Jose, Ch. 6.

29. Robert B. Kelley, "Binary and Gray Scale Robot Vision", *Robotics and Robot Sensing Systems, Proceedings of SPIE-The International society for Optical Engineering*, Vol. 442, CA: San Diego, August, 1983, pp. 27-37.

30. Mohan M. Trivedi, Chuxin Chen, and Suresh B. Marapane, "A Vision System for Robotics Inspection and Manipulation", *Computer*, June, 1989, pp. 91-96.

31. Robert W. Fox and Alan T. Mcdonald, "Bernoulli Equation", *Introduction to Fluid Mechanics*, 4th edition, New York: John Wiley & Sons, Inc. 1992.

32. Forrest M. Mims, "Engineers Mini-Notebook Science Projects", *Siliconcepts Book*, 3rd Printing, 1994, p.36.

33. Marsh W. White, Kenneth V. Manning, and Robert L. Weber, "Magnetic Effects of Electric Current", *Basic Physics*, NY: McGraw-Hill Book Company, 1968, Ch. 23.

34. Frank L. Robeson, "Magnetism", *Physics*, New York: The Macmillan Company, 1942, Cp. 15.

35. Douglas C. Giancoli, "Source of Magnetic Field", *Physics for Scientists and Engineers*, 2nd edition, New Jersey: Prentice Hall, 1988, Ch. 30, Sec. 6.

36. G. Lindfield and J. Penny, "Fitting Function to Data", *Numerical Methods Using MATLAB*, Great Britain: Ellis Horwood Limited, 1995, Ch. 7, Sec. 6.

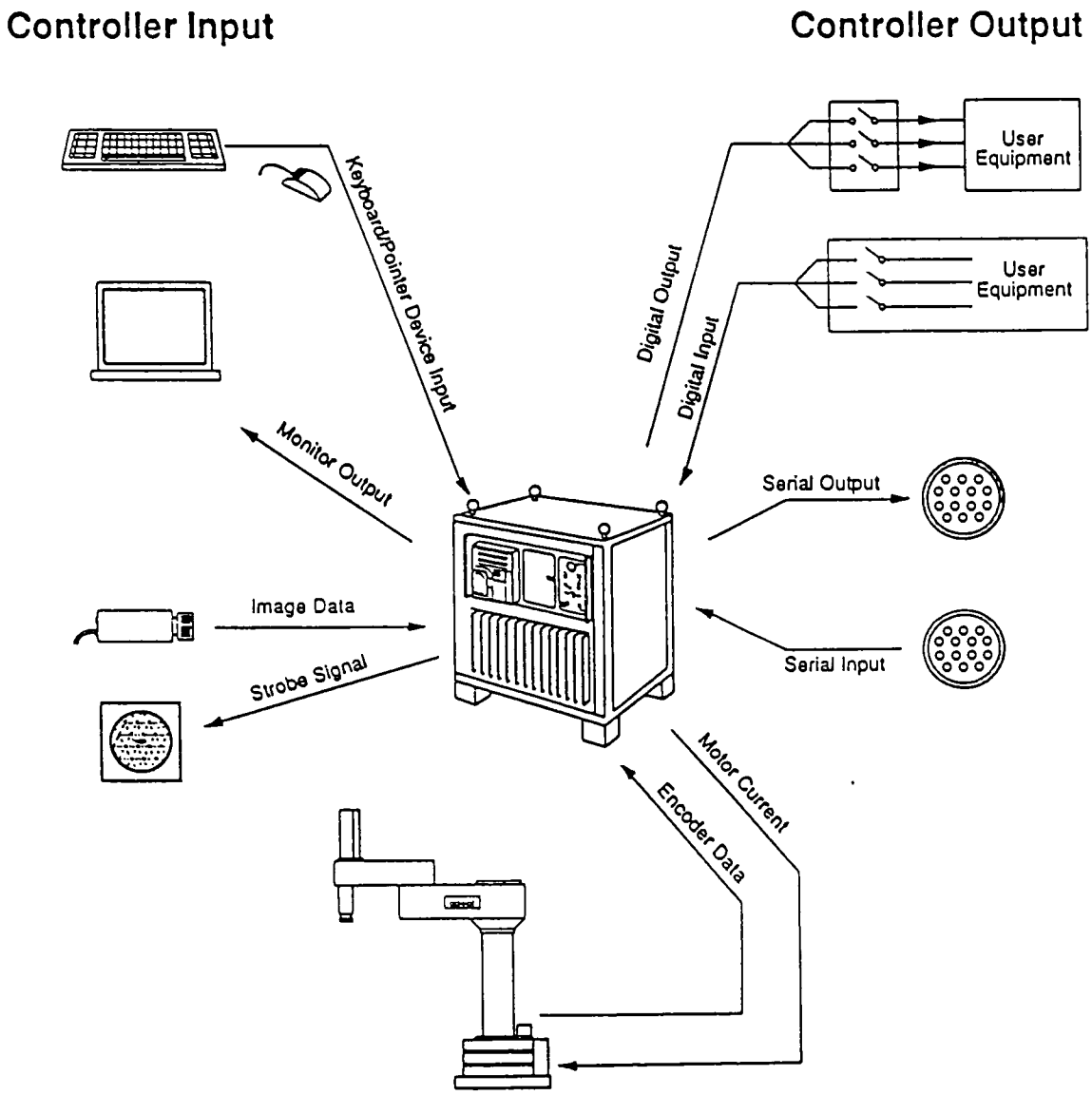
37. MATLAB is a registered trademark of The Mathworks, Inc., 24 Prime Park Way, Natick, MA, USA, 01760-1500.

38. John E. Freund and Gary A. Simon, "The Coefficient of Correlation", *Modern Elementary Statistics*, 8th edition, New Jersey: Prentice Hall, 1992, Ch. 16.

39. Kenneth V. Manning, Marsh W. White, and Robert L. Weber, "Light and Illumination", *Basic Physics*, NY: McGraw-Hill Book Company, 1968, Ch. 2.

40. Kenneth W. Tobin Jr., "Machine-Vision Expertise Can Lead to Industrial Success", *Laser Focus World*, September, 1994, pp.115-119.

Appendix A *A Typical AGS-GV System* [28]



Keyboard/Pointer Device Input --- The keyboard and the pointer are used for data or command input. A command can be entered by typing to control the operation of the robot under basic V/V+ environment. When motion software or vision software is used, the value inside the record can also be entered by these input devices to establish each database.

Monitor --- The monitor is a basic component, used to show the controller response and the vision image. If the vision system is operating: the image representation, vision tool location, and program execution can be seen from the monitor. Resolution of the monitor is an important factor which affects the vision inspection results because the scales of the vision tools require adjustment in order to acquire the correct image during inspection.

Camera --- The camera is used to provide the vision ability of the robot system. The aperture number and focusing distance of a camera can be adjusted to generate good object images. The pixel number of the image surface inside a camera can affect the resolution of the vision system. A proper resolution is required for different vision operations.

Strobe Light --- The strobe lighting source is an optional piece of equipment for the robot system. It is controlled by the strobe signal to provide the light impulse that illuminates the object.

Serial Input/Output Device --- The I/O device is located behind the robot controller. It is utilized to connect the signal cables between the controller and the user's equipment. There are eight available connectors integrated in the serial device.

Digital Input/Output --- The signal used to control the user's equipment is generated by the digital I/O device. It makes use of binary strategy, on and off, for signal sending or receiving. The controller can identify the signal to control the robot motion or in order to actuate other equipment.

Appendix B *MotionWare Commands and Their Functions*

1. General MotionWare Statements

MOVE

Move is a common command used to control the motion of a robot. It indicates for a robot to move to a pretaught location or a position which is located in a relative reference frame.

SET_FRAME

A vision command. It is used to set a relative reference frame, whose center coincides with the image center. The translation or rotation of a part can be specified by this reference frame. The processor is then able to guide an end effector to that location.

SET_TOOL

If a special tool is integrated to the robot system, an off-set may be needed. This command is used to calculate each off-set. After tool records have been created, the multiple off-sets can be calculated by the controller applying to the exact locations. The robot moves tools to the desired positions, rather than moving its arm to those locations.

2. MotionWare Control Structures

IF

A logic command. It executes a block of statements based on the status of the digital signals. This command can also be used to check the vision executions.

THEN

A conditional statement following an IF statement, which executes the block of IF statements, if and only if the conditions of the IF statement are successful.

ELSE

This is always used in conjunction with IF and THEN commands. It executes another sub-block statement if a fault has occurred in the IF statement.

END

The END command is used to terminate the conditional statements. When editing a sequence, the END command must be the matched pair of an IF statement.

IO

The IO statement is used by soft digital signals to control the other equipment of a robot system. It can combine with IF statements for better robot control based on the signals setting.

WAIT_FOR

Waiting for a particular signal to be correctly set before executing the next statement.

WAIT_UNTIL

This command stop the robot or the next statement execution until digital signals are correctly set. Only when those signals are actuated, will the robot then be able to execute the next block of statements.

REPEAT

The REPEAT command allows a block of statements to be executed a specified number of times. This command can abbreviate the length of a program.

WHILE

When using this WHILE command, a block of statements will definitely execute as long as the specified digital signals are correctly set. This command provides capabilities for controlling the robot behavior by signal inputs.

NEXT_LOOP

This statement executes a portion of the loop block when certain conditions are correct. It usually contains IF and END statements that identify a condition which indicates that the loop should be repeated.

EXIT_LOOP

A termination command used to stop a REPEAT or WHILE loop, while ordering the program to continue at the next statement following the END statement that closes the loop.

3. Conveyor Statements

ATTACH_FRAME

A special statement used with moving reference frames. A specified frame will attach to the center of a moving part and will decide its moving speed and location. The robot increases or decreases its speed to match the speed of this part.

CHECK_FRAME

CHECK_FRAME is used in conjunction with ATTACH_FRAME. It checks whether the parts being used in the ATTACH_FRAME statement are obtainable or not. If the parts can not be obtained, the robot will wait until parts enter into an obtainable distance.

STOP_ROBOT

It is used to stop the robot for a specific time when an error condition is detected during a belt tracking motion. An optional argument is incorporated with this statement for indicating stop the robot at the end of the current motion or abort immediately.

4. MotionWare Statements Requiring Vision

LOCATE_ASSEMBLY

This particular command is applied to create a reference frame for an assembly. By inspecting some specified fiducial marks within the assembly area, LOCATE_ASSEMBLY uses vision to determine the precise value of the reference frame, then it updates the location database associated with the new value of that reference frame.

MOVE_CAMERA

When using an arm-mounted camera to take pictures of an indexing conveyor, the robot must move before vision data can be generated. In this case, MOVE_CAME generates a robot motion when new picture data is required.

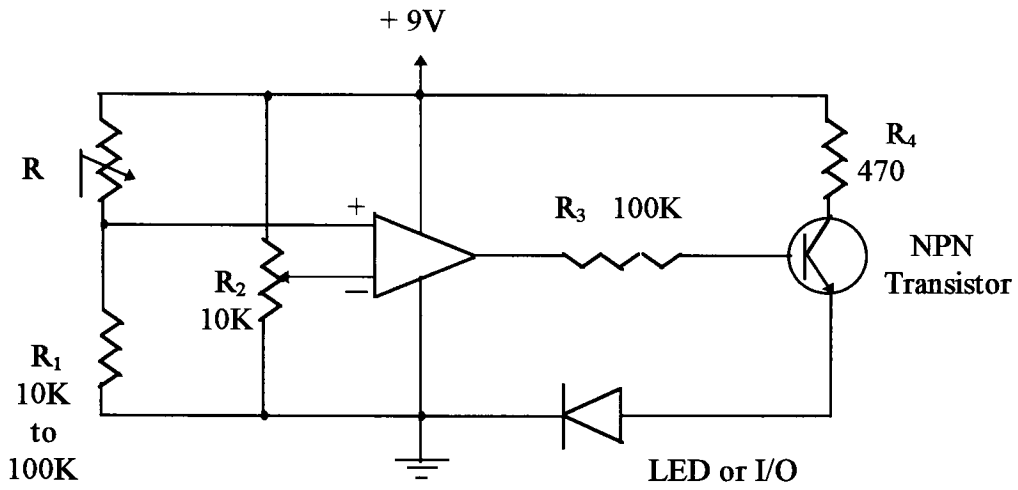
MOVE_INSPECT

Moves the camera to specified locations to apply vision tools on these positions. The processor determines the inspections' result on those locations, then controls the motion of the robot. The robot will then execute the next statement unless a pass signal is generated by the MOVE_INSPECT command.

MOVE_REFINE

This statement is used for an upward facing camera. The robot moves the gripper in front of this camera to process vision inspection and calculates the part reference frame for the gripped part.

Appendix C *Force Sensing Circuit*



R --- R is a force sensor variable resistor.
R_1 --- R_1 is the resistor used to reduce the current across the resistor R.
R_2 --- R_2 is a variable resistor used to establish the threshold voltage.
R_3, R_4 --- R_3 and R_4 two resistors are utilized to reduce the voltage entering to the NPN transistor.

NPN Transistor --- The NPN transistor works as a switch. The collector is the 9V power supply, emitter is a signal, and base is the ground. The actuation of the NPN transistor is determined by the signal output from the OP-AMP. A high signal output from the OP-AMP will actuate the NPN transistor so that the LED is turned on, and vice versa.

OP-AMP --- The operational amplifier is a very high gain dc-coupled differential amplifier, with single-end outputs. The output goes positive when the noninverting input (+) becomes more positive than the inverting input (-), and vice versa.

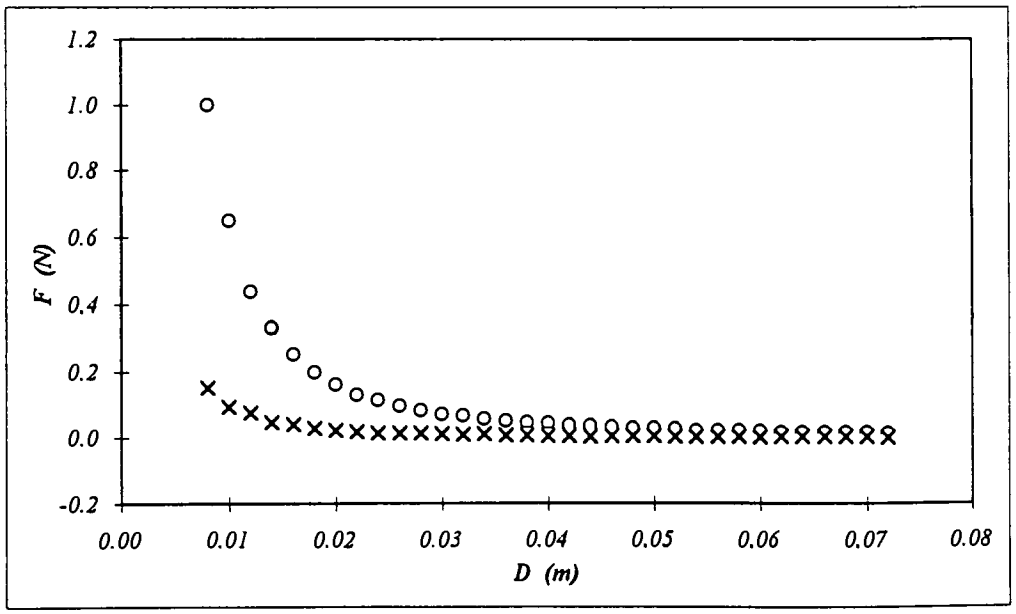
LED or I/O --- The LED is used to indicate signal actuation. The LED turning on means that there is an output signal actuated, and vice versa. Instead of using an LED, this circuit can connect the other equipment for signal input and output usage.

This force sensing circuit operates as follows: When the force or pressure acting on the resistor R_1, voltages of the noninverting (+) and inverting (-) are different. If '+' voltage is higher than the '-' voltage, an output voltage will actuate the NPN transistor such that there is a current crossing the LED. This current is detected by the robot controller as an input signal. If '+' voltage is lower than the '-' voltage, the NPN transistor will not be actuated: so there is no signal detected. Resistor R in the circuit is assigned to series connected potentiometers in the end effector design. Resistor R_2 is an external adjustable resistor used to establish the threshold voltage. If the force achieves the desired value, resistor R_2 is adjusted to balance voltage differences until the LED turns off. In this manner, the threshold voltage corresponding to the correct force acting on the chip is obtained. If there is another force applied on the chip, this threshold voltage will be used in comparison with the current voltage. The robot system will then generate a digital signal if this voltage is higher than the threshold voltage, and vice versa. By identifying the generated signal, behavior of the end effector and the vacuum actuation can be controlled.

Appendix D *Magnetic Force*

D : Distance between two magnetic poles SS : Force of two SS poles*
NN : Force of two NN poles**

D (mm)	Force (N)		D (mm)	Force (N)	
	SS	NN		SS	NN
0.000	N/A	N/A	0.040	0.0053	0.0412
0.008	0.1506	0.9987	0.042	0.0049	0.0371
0.010	0.0923	0.6510	0.044	0.0045	0.0331
0.012	0.0755	0.4389	0.046	0.0042	0.0320
0.014	0.0459	0.3301	0.048	0.0040	0.0278
0.016	0.0401	0.2510	0.050	0.0037	0.0255
0.018	0.0269	0.1980	0.052	0.0033	0.0211
0.020	0.0225	0.1602	0.054	0.0032	0.0199
0.022	0.0186	0.1299	0.056	0.0029	0.0201
0.024	0.0149	0.1123	0.058	0.0028	0.0192
0.026	0.0133	0.0947	0.060	0.0026	0.0180
0.028	0.0112	0.0818	0.062	0.0023	0.0168
0.030	0.0098	0.0709	0.064	0.0024	0.0152
0.032	0.0106	0.0654	0.066	0.0022	0.0147
0.034	0.0096	0.0562	0.068	0.0019	0.0140
0.036	0.0072	0.0506	0.070	0.0016	0.0122
0.038	0.0062	0.0443	0.072	0.0013	0.0119



o : NN pair x : SS pair

*,** : Mean Value

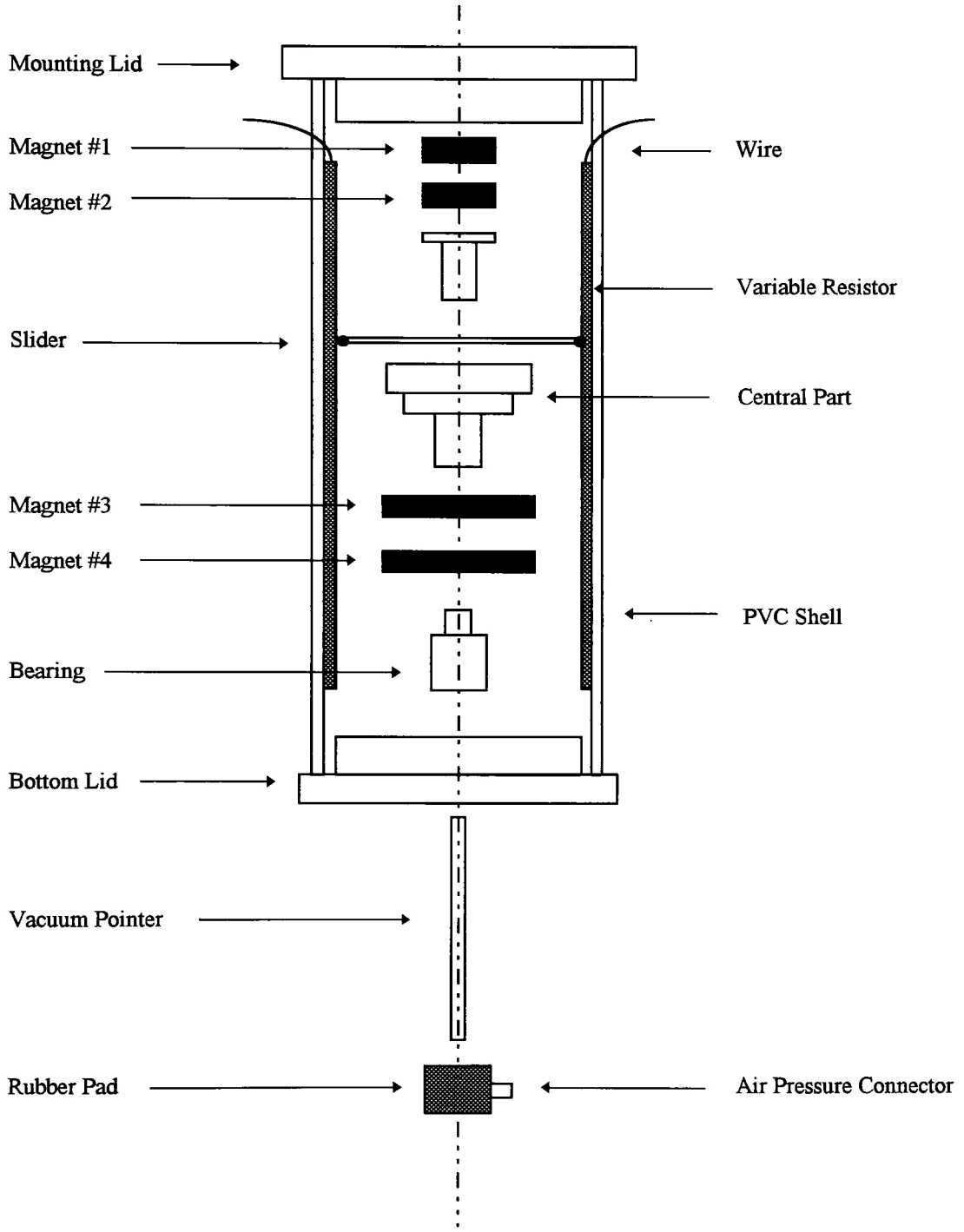
Appendix E Data of Actual Magnetic Force and Simplified Magnetic Force

D : Robot moving distance
Fmg : Weight of the central part
Frepel : Repulsive magnet forces
Fcalnet : Calculated total magnetic net forces

Fexpt. : Measured experimental values
Fnet : Experiment magnetic net forces
Fatt : Attractive magnet forces
Fsimpl : Simplified magnetic net forces (based on 2nd & 3rd terms)

D	F expt.	F mg	F net	F repel	F att	F calnet	F simpl.
0	0.0000	0.2253	-0.2253	0.63850	-0.8412	-0.20270	-0.24000
0.001	0.0219	0.2253	-0.2034	0.61655	-0.8047	-0.18815	-0.21075
0.002	0.0455	0.2253	-0.1798	0.61055	-0.7740	-0.16345	-0.18605
0.003	0.0585	0.2253	-0.1668	0.60683	-0.7556	-0.14877	-0.17137
0.004	0.0726	0.2253	-0.1527	0.60576	-0.7481	-0.14234	-0.16494
0.005	0.0872	0.2253	-0.1381	0.60201	-0.7261	-0.12409	-0.14669
0.006	0.0951	0.2253	-0.1302	0.59938	-0.7075	-0.10812	-0.13072
0.007	0.1102	0.2253	-0.1151	0.59769	-0.6917	-0.09401	-0.11661
0.008	0.1292	0.2253	-0.0961	0.59713	-0.6785	-0.08137	-0.10397
0.009	0.1362	0.2253	-0.0891	0.59748	-0.6674	-0.06992	-0.09252
0.01	0.1484	0.2253	-0.0769	0.59911	-0.6585	-0.05939	-0.08199
0.011	0.1554	0.2253	-0.0699	0.60173	-0.6513	-0.04957	-0.07217
0.012	0.1668	0.2253	-0.0585	0.60574	-0.6460	-0.04026	-0.06286
0.013	0.1751	0.2253	-0.0502	0.61115	-0.6424	-0.03125	-0.05385
0.014	0.1806	0.2253	-0.0447	0.61824	-0.6406	-0.02236	-0.04496
0.015	0.1959	0.2253	-0.0294	0.62735	-0.6407	-0.01335	-0.03595
0.016	0.2011	0.2253	-0.0242	0.63870	-0.6427	-0.00400	-0.02660
0.017	0.2236	0.2253	-0.0017	0.65302	-0.6470	0.00602	-0.01658
0.018	0.2203	0.2253	-0.0050	0.67088	-0.6538	0.01708	-0.00552
0.019	0.2544	0.2253	0.0291	0.69354	-0.6638	0.02974	0.00714
0.02	0.2576	0.2253	0.0323	0.72244	-0.6777	0.04474	0.02214
0.021	0.2832	0.2253	0.0579	0.76002	-0.6968	0.06322	0.04062
0.022	0.3050	0.2253	0.0797	0.80996	-0.7230	0.08696	0.06436
0.023	0.3279	0.2253	0.1026	0.87820	-0.7593	0.11890	0.09630
0.024	0.3502	0.2253	0.1249	0.97550	-0.8113	0.16420	0.14160

Appendix F *Cross-Section Schematic of the End Effector*



Appendix G *Calculation of Polynomial Coefficients Using MATLAB*

Coefficient Calculations of Polynomial Fitting Function

This code is used to calculate the coefficients of various fitting function. The experimental data of forces, voltages, and distance are saved as force.m, voltage.m, and distance.m. The coefficients of fitting function are generated by the command **polyfit**, and the results are printed. The original data and the fitting functions are plotted.

```
clear
   load distance.m; load force.m; load voltage.m;
   x=[0:0.1:24];xx=[0:0.01:0.4];
   p1=polyfit(distance,voltage,1), y1=polyval(p1,x);
   p2=polyfit(distance,force,3),   y2=polyval(p2,x);
   p3=polyfit(force,voltage,2),    y3=polyval(p3,xx);

   figure(1);
      plot(distance,voltage,'o',x,y1); axis([0 25 0 8]);
      xlabel('Distance  (mm)'); ylabel('Voltage  (volts)');

   figure(2);
      plot(distance,force,'o',x,y2); axis([0 25 0 .4]);
      xlabel('Distance  (mm)'); ylabel('Force  (N)');

   figure(3);
      plot(force,voltage,'o',xx,y3); axis([0 .4 0 8]);
      xlabel('Force  (N)'); ylabel('Voltage  (volts)');
```

* * * * * * * * * * * *

```
   P1
           -6.75e001          0.6772e001

   P2
            3.8644e-005     -1.316e-003     2.3915e-002     -1.3998e-003

   P3
            0.28086e001     -0.6185e001     0.69017e001
```

Appendix H Data of End Effector Calibration

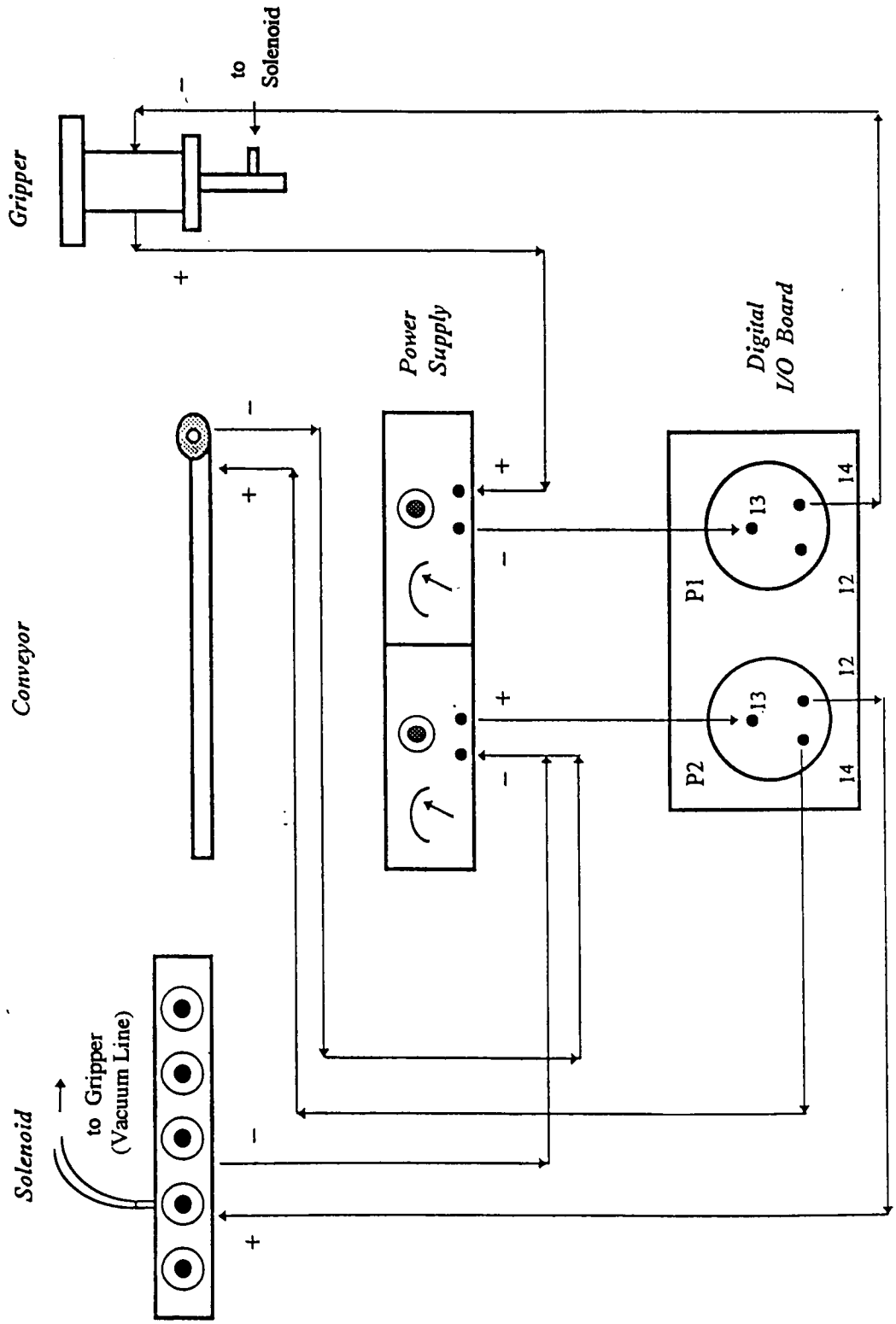
F1-F5 : 5 measurements of magnetic forces
Favg : Average magnetic forces
FSD : Standard deviation of force measurement

V1-V5 : 5 measurements of output voltages
Vavg : Average output voltages
VSD : Standard deviation of voltage measurement

mm \ N	F1	F2	F3	F4	F5	Favg	FSD
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.0286	0.0242	0.0258	0.0194	0.0113	0.0219	0.0068
2	0.0467	0.0471	0.0464	0.0412	0.0463	0.0455	0.0024
3	0.0644	0.0628	0.0456	0.0602	0.0595	0.0585	0.0075
4	0.0722	0.0728	0.0818	0.0679	0.0681	0.0726	0.0056
5	0.0834	0.0898	0.0871	0.0948	0.0808	0.0872	0.0055
6	0.0983	0.0950	0.0960	0.0953	0.0911	0.0951	0.0026
7	0.1099	0.1093	0.1088	0.1151	0.1081	0.1102	0.0028
8	0.1275	0.1355	0.1289	0.1243	0.1296	0.1292	0.0041
9	0.1388	0.1329	0.1365	0.1344	0.1385	0.1362	0.0026
10	0.1422	0.1487	0.1464	0.1495	0.1552	0.1484	0.0047
11	0.1523	0.1522	0.1536	0.1567	0.1622	0.1554	0.0042
12	0.1673	0.1576	0.1697	0.1707	0.1687	0.1668	0.0053
13	0.1733	0.1765	0.1801	0.1755	0.1699	0.1751	0.0038
14	0.1882	0.1805	0.1818	0.1808	0.1718	0.1806	0.0058
15	0.1960	0.1899	0.2003	0.1955	0.1978	0.1959	0.0038
16	0.2037	0.1911	0.2082	0.2047	0.1979	0.2011	0.0067
17	0.2285	0.2237	0.2174	0.2239	0.2243	0.2236	0.0040
18	0.2338	0.2442	0.2238	0.1989	0.2006	0.2203	0.0201
19	0.2547	0.2599	0.2598	0.2551	0.2424	0.2544	0.0071
20	0.2665	0.2599	0.2466	0.2544	0.2605	0.2576	0.0075
21	0.2966	0.2873	0.2858	0.2655	0.2806	0.2832	0.0114
22	0.3131	0.3079	0.3056	0.2970	0.3014	0.3050	0.0062
23	0.3313	0.3300	0.3119	0.3099	0.3562	0.3279	0.0187
24	0.3536	0.3549	0.3481	0.3483	0.3460	0.3502	0.0039

mm	V1	V2	V3	V4	V5	Vavg	VSD
0	6.84	6.84	6.84	6.84	6.84	6.84	0.0000
1	6.72	6.72	6.73	6.72	6.75	6.73	0.0134
2	6.64	6.63	6.62	6.61	6.60	6.62	0.0153
3	6.59	6.53	6.55	6.54	6.55	6.55	0.0208
4	6.44	6.43	6.47	6.45	6.39	6.44	0.0297
5	6.36	6.38	6.32	6.39	6.35	6.36	0.0274
6	6.29	6.30	6.28	6.28	6.29	6.29	0.0084
7	6.22	6.23	6.19	6.20	6.21	6.21	0.0158
8	6.14	6.14	6.15	6.12	6.15	6.14	0.0122
9	6.09	6.11	6.10	6.08	6.12	6.10	0.0158
10	6.03	6.05	6.06	6.04	6.03	6.04	0.0130
11	5.99	5.98	5.92	6.01	6.00	5.98	0.0354
12	5.92	5.96	5.91	5.97	5.96	5.94	0.0270
13	5.92	5.89	5.86	5.91	5.88	5.89	0.0239
14	5.86	5.87	5.85	5.88	5.86	5.86	0.0117
15	5.80	5.79	5.78	5.81	5.88	5.81	0.0396
16	5.74	5.74	5.72	5.72	5.73	5.73	0.0101
17	5.59	5.65	5.71	5.67	5.66	5.66	0.0434
18	5.58	5.58	5.59	5.60	5.59	5.59	0.0090
19	5.56	5.57	5.53	5.52	5.57	5.55	0.0235
20	5.46	5.44	5.44	5.45	5.42	5.44	0.0151
21	5.38	5.12	5.36	5.63	5.52	5.40	0.1921
22	5.26	5.27	5.28	5.27	5.28	5.27	0.0071
23	5.17	5.15	5.15	5.16	5.16	5.16	0.0058
24	4.98	4.89	4.99	5.10	5.06	5.00	0.0808

Appendix I *Schematic of a Complete Circuit Connection*



Appendix J *Precision Determination*

J - 1 Precision Determination for DIL Insertion

In this section, the precision determination of the *DIL* insertion is described. The dimensions of the *DIL*, *SMD*, and *PCB* are listed in *Table 1* and *2* in *Chapter II*. For a *DIL*, the width of pin tip is *0.5mm*, pin thickness is *0.2mm*, and the diameter of the inserting hole is *1mm*. Considering these values, the assembly precision can be calculated. In order for successful insertion, the maximum shifting in the *X* and *Y* direction, and the acceptable rotation angles, must be measured.

Translation Precision — *Figure J-1.1* shows the top view of a correct insertion and the maximum shifting distance in both *X* and *Y* directions.

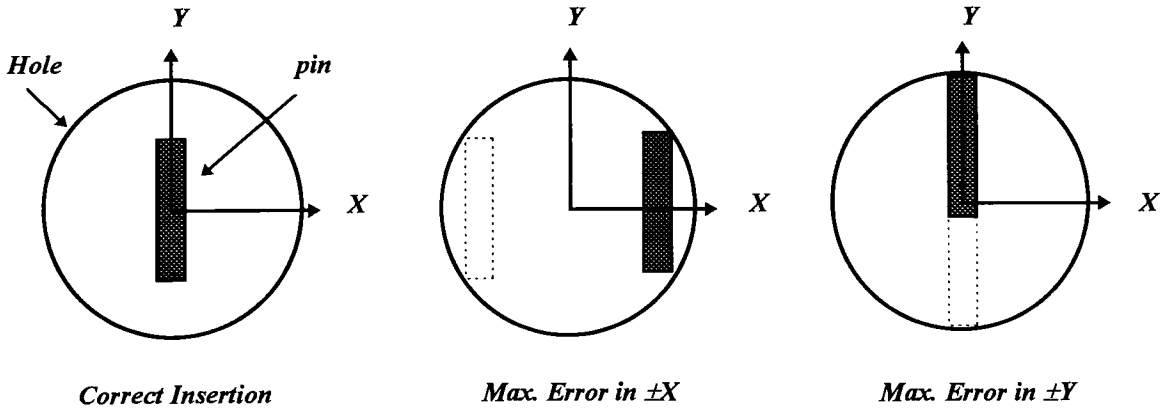
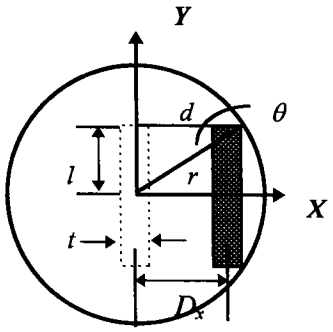


Figure J-1.1. Pin Shifted Direction of a *DIL* After Insertion

From the above figure, maximum translation error in *X* and *Y* directions can be simply decided from the following calculations.

Max. Error in ±X Direction:



where

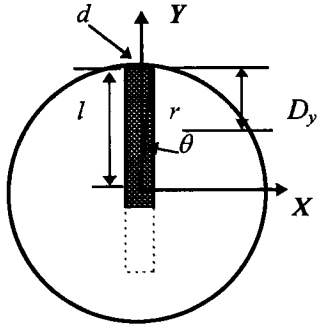
l = half of the pin length
d = horizontal distance from Y axis to the hole edge
r = radius of the hole
θ = angle between d and r
t = width of the pin
Dx = max. error in the X direction

$r\,Sin\theta = l \Rightarrow 0.5\,Sin\theta = 0.25 \Rightarrow \theta = 30^{o}$

$d = r\,Cos\theta = 0.5\,Cos30^{o} = 0.433mm$

Max. Error in the +X direction is: $D_x = d - t/2 = 0.433 - 0.1 = 0.333mm$

Max. Error in ±Y Direction:



where

l = length from X axis to the center of pin
r = distance from origin to the corner of pin
θ = angle between r and l
d = distance between Y axis and the corner
Dy = max. error in the Y direction

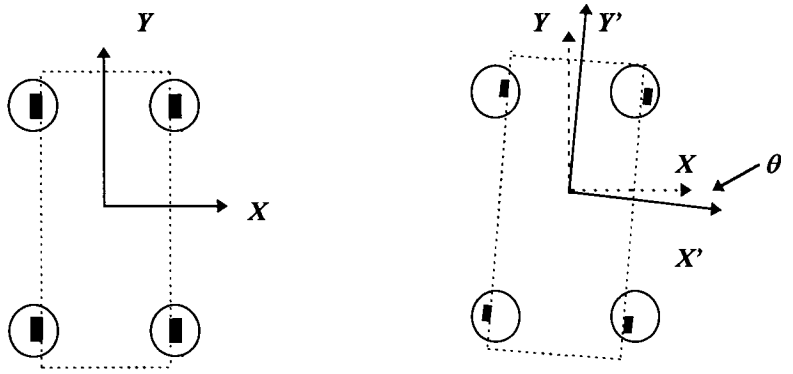
$r\,Sin\theta = d \Rightarrow 0.5\,Sin\theta = 0.1 \Rightarrow \theta = 11.537^{o}$

$l = r\,Cos\theta = 0.5\,Cos11.537 = 0.49mm$

Max. Error in the +Y direction is: $D_y = l - 0.25 = 0.24mm$

From the above calculations, the maximum shifting distance in *X* and *Y* directions are *±0.33mm* and *±0.24mm*, respectively. The last digit number is truncated in order not to achieve the limitations. Therefore, the translation precision requirements for a *DIL* assembly are: *±0.3mm* in *X* direction and *±0.2mm* in *Y* direction.

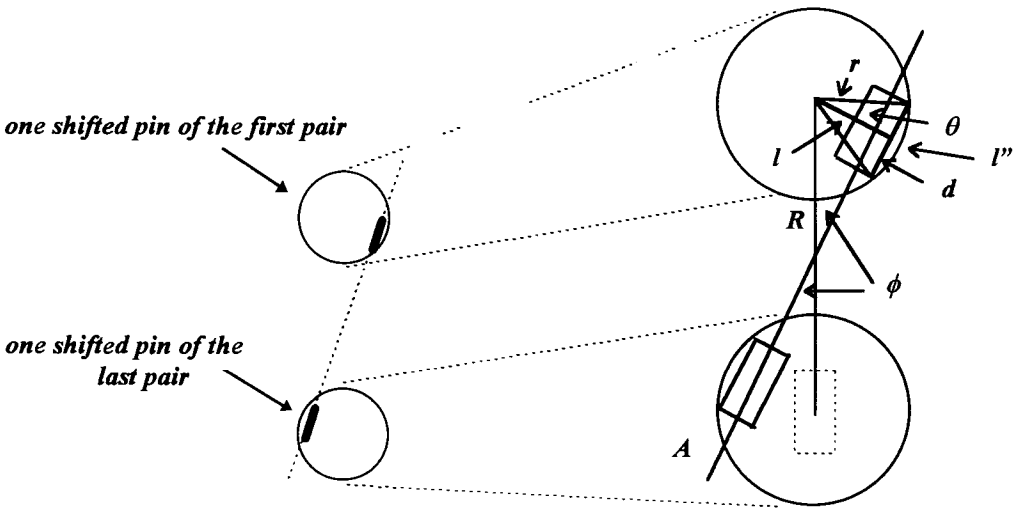
Rotation Precision — After translation precision has been determined, the next consideration is the rotational accuracy requirement. The typical length and width for a *DIL* are *19mm* and *8mm*. Based on the data, rotational precision can be decided. Since the *DILs* used in this project are of symmetric geometry, it is only necessary to consider four pins: the first and the last pairs. The maximum rotational error can be decided from these two pairs and theirs distance apart from each other. There are two situations that can be used to calculate rotation precision: the first situation is when one pin is located at the center, but the other three pins have shifted, the second situation is when all four pins are not located at the center. The first situation gives a smaller rotational angle than the second one, so the second case is used to calculate rotational inaccuracy. *Figure J-1.2* shows the top view of the first and the last pair of the pins after insertion.



Correct Insertion *Insertion with rotation error*

Figure J-1.2. Rotation Error After Insertion

Max. Rotation Error:



where

A = a line connects the center of two shifted pins

R = half length between the center of first hole to the center of last hole

r = hole radius

d = half length of the pin width

l = distance from the hole center to the pin edge

l" = arc length

θ = angle between r and l

ϕ = angle between line R and line A

From the above enlarged figure:

$r\,2\theta = l''$ *Since* θ *is very small* $\Rightarrow$ $l'' \approx 2d \Rightarrow r\,2\theta = 2d$

$0.5 \bullet 2\theta = 0.5 \Rightarrow \theta = 0.5$ *rad.* $\Rightarrow$ $l = r\,Cos\theta = 0.5\,Cos0.5 = 0.438mm$

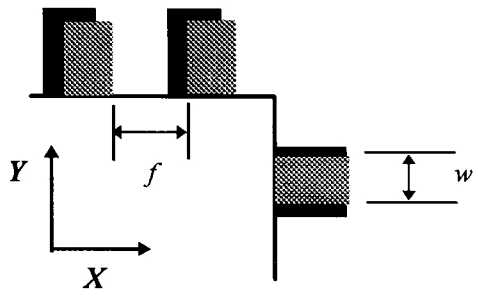
Max. rotational error is:

$R\phi = l - w/2$ *(w is the width of the pin)* $\Rightarrow$ $19/2\,\phi = 0.438 - 0.1 = 0.338$

$\Rightarrow \phi = 0.035578$ *rad* $= 2.038^{o}$

J - 2 Precision Determination for SMD Insertion

Translation Error in $\pm X$ Direction:



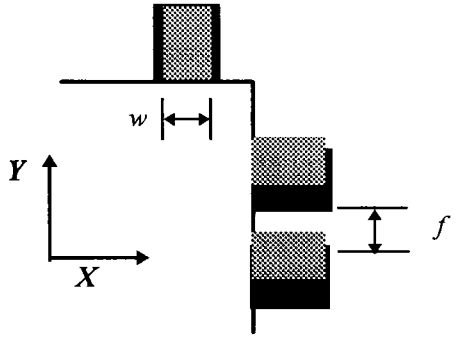
Pin width $= w = 0.5mm$

Footprints space $= f = 0.5mm$

Max. allowable shifting distance in the $\pm X$ *direction is:*

$D_x = f = \pm 0.5mm$

Translation Error in $\pm Y$ Direction:



Pin width $= w = 0.5mm$

Footprint space $= f = 0.5mm$

Max. allowable shifting distance in the $\pm Y$ *direction is:*

$D_y = f = \pm 0.5mm$

From the above figures, the maximum translation tolerances in the X and Y directions are $0.5mm$. This value is not valid for the assembly because pins can not be located on the footprints' spaces. Therefore, another acceptable precision requirement is established: the maximum translation error must not exceed half the width of the pins. The reason for setting this restricted accuracy is that the adhesive materials used to solder a *SMD* will be squeezed when the *SMD* locates on the footprints. If this case happens, the invalid contact between two pins occurs. Based on this consideration, the maximum acceptable translation accuracy is $\pm 0.25mm$ in both X and Y directions.

Rotation Precision — A *SMD* orientation error can be decided by using two radii: the width and the length of *SMD*. Since the *SMD* is a rigid body, using width or length as its rotating radius provides the same rotational angle. Although the rotating angle is the same, using the length as the radius gives a larger rotating distance. Therefore, the maximum acceptable rotational accuracy must be calculated by using the length of the *SMD* as the rotation radius. It also should be noted that the rotational accuracy is only considered to be pure rotation, without any translation in *X* or *Y* direction. The following figure shows the rotation error.

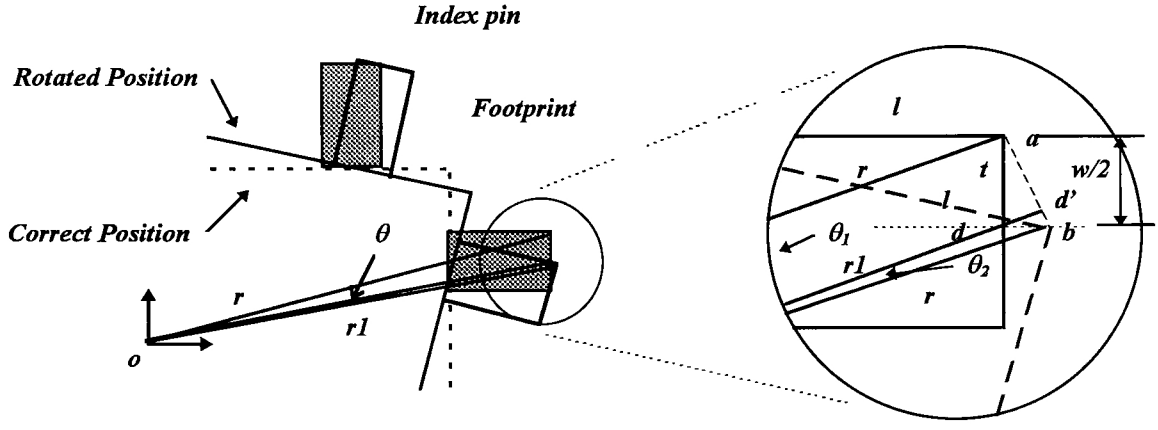


Figure J-2.1 Rotation Error for *SMD* Placement

where

o = rotation center (center of SMD)
l = length of the pin
a = corner point before rotation
b = corner point after rotation
d = middle point of the pin
d' = projection distance from point d to line ab
t = half length of the pin width
w = width of the pin (t = w/2)
θ1 = angle between line oa and od'
θ2 = angle between line od' and ob
r = rotation radius (from SMD center to the corner of one index pin)
r1 = length from rotation center to the middle point of the pin

The rotating radius r is equal to 13mm, r1 is equal to 12.85mm, and w/2 is equal to 0.25mm. By applying the Cosine Law, θ_1 can be found

$$Cos\,\theta_1 = \frac{r^2 + r_1^2 - t^2}{2 \cdot r \cdot r_1} = \frac{13^2 + 12.85^2 - 0.25^2}{2 \cdot 13 \cdot 12.85}$$

$\Rightarrow \quad \theta_1 = 0.8866^o \quad \Rightarrow \quad$ *Since* θ_1 *is very small,* $r\,\theta_1$ = *arc length* $\approx ad'$

$\Rightarrow \quad ad' = 13 \cdot 0.8866 \cdot \frac{\pi}{180} = 0.201mm$

The angles inside $\Delta add'$ *can be decided by using the length ad and dd'*
$ad = t = w/2 = 0.5 / 2 = 0.25mm.$ *and* $dd' = r - r_1 = 13 - 12.85 = 0.15mm$

$$Cos\angle add' = \frac{0.25^2 + 0.15^2 - 0.2^2}{2 \cdot 0.25 \cdot 0.15} \quad \Rightarrow \quad \angle add' = 53.13^o$$

$$Cos\angle dad' = \frac{0.25^2 + 0.2^2 - 0.15^2}{2 \cdot 0.25 \cdot 0.2} \quad \Rightarrow \quad \angle dad' = 36.87^o$$

So $\angle dd'a = 180^o - \angle dad' \quad \angle add' = 180 - 90^o = 90^o$
Also $\angle bd'd = 180^o - \angle dd'a = 180^o - 90^o = 90^o$
Thus $\angle bdd' = 36.87^o$ *and* $\angle bd'd = 53.13^o$
Apply Sine Low to $\Delta bdd'$

$$\frac{bd}{Sin90^o} = \frac{bd'}{Sin36.87^o} = \frac{dd'}{Sin53.13^o}$$

$$\Rightarrow \quad bd = \frac{0.15}{0.8} = 0.1875mm \quad and \quad bd' = \frac{0.15}{0.6} = 0.1125mm$$

Therefore, the total arc length A due to a rotational error is

$A = ad' + bd' = 0.201 + 0.1125 = 0.3135mm$

Thus, the maximum acceptable rotating angle can be found:

$$A = r\,\theta \quad \Rightarrow \quad 0.3135 = 13\,\theta \quad \Rightarrow \quad \theta = 1.3817^o$$

From the above calculations, the rotational precision requirement for a *SMD* assembly is $\pm 1.3817^o$ about its center.

J - 3 Mirror Inspection Precision

This section describes the precision requirement when a *DIL* is placed in front of the mirror. This mirror is used to check the pin conditions before introducing a *DIL* to the *PCB*. The mirror is experimentally set up at *60* degrees such that the pin image can be totally reflected into

the camera. Since the basic precision requirements have been calculated in section *J-1*, a mirror transformation is the remaining task for a *DIL* inspection. The basic accuracy requirements is to locate a *DIL* within *±0.3mm* in the *X* direction and *±0.2mm* in the *Y* direction. This accuracy can also be viewed in another way: the pin of a *DIL* can have *±0.3mm* and *±0.2mm* tolerances in *X* and *Y* directions if there is no insertion error. This assumption gives a fundamental inspection precision when checking the pin conditions. *Figure J-3.1* shows the pin errors in the *X*, *Y*, and *Z* direction, and *Figure J-3.2* shows the positions of a mirror and a lifted *DIL* presenting in front of the mirror. The pin image inspection is described as follows.

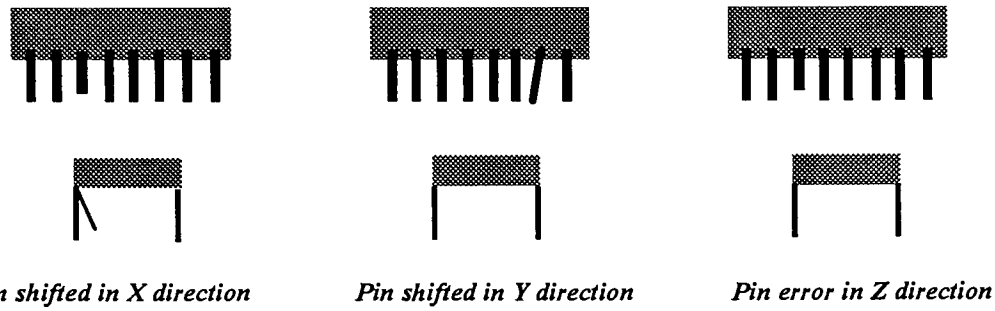


Figure J-3.1 Pin Shifted or Error in X, Y, and Z Directions

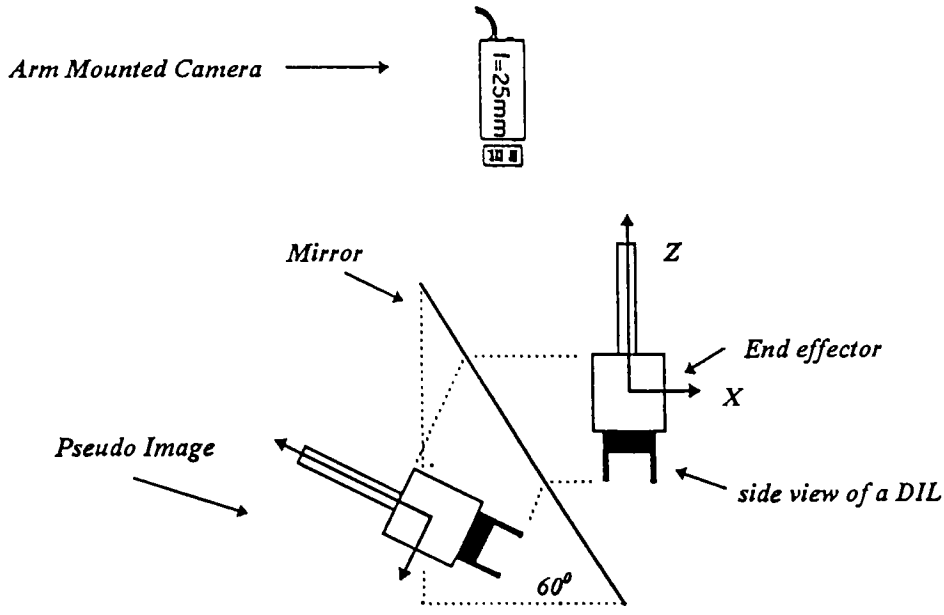


Figure J-3.2 Mirror Set-up and DIL Location

Pin Shifted in ±Y Direction — From *Figure J-3.1*, the *±Y* direction is parallel to the mirror surface, so no image transformation has occurred. One unit shifting distance corresponds to one unit image shifting distance. Therefore, the precision requirement for pin alignment in *±Y* direction is *±0.2mm*.

Pin Shifted in ±X Direction — In the *X* direction, the pin misalignment is not parallel to the mirror surface. The mirror transformation should be employed in this direction. When there is a pin misalignment, the distance in between pin tip and the mirror is smaller or larger than the normal position; and the pin pitch also changes. By inspecting the distance and pin pitch, the pin alignment can be determined. *Figure J-3.3* shows the mirror transformation, and the precision requirement for the mirror inspection in *X* direction is calculated as follows.

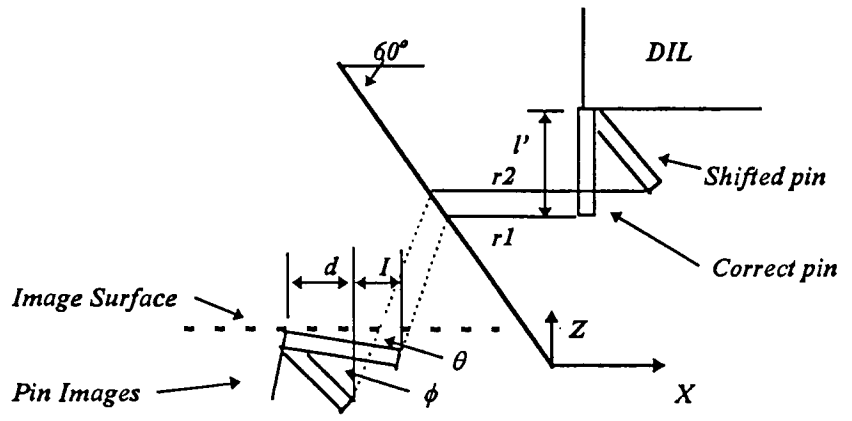


Figure J-3.3 Mirror Transformation

where

r1 = correct distance from the pin tip to the mirror
r2 = shifted distance from the pin tip to the mirror
d = shifted pin length on the image surface
l = length difference between the correct pin and shifted pin on the image surface
l' = pin length
θ = angle between the correct pin image and image surface
ϕ = angle between the shifted pin image and image surface

From the above figure and the X direction shifting distance 0.3mm, r_2 can be calculated:

$$r_2 = r_1 + 0.3 = 30 + 0.3 = 30.3mm$$

Using the mirror symmetric characteristic and simple triangular relationships, the angle θ can be obtained.

$$\Rightarrow \quad \theta = 30^{o}$$

In addition, the angle ϕ is the same as a real angle, so ϕ can be calculated from the following:

$$l' \, Sin\phi = 0.3 \quad \Rightarrow \quad 6.5 \, Sin\phi = 0.3 \quad \Rightarrow \quad \phi = 2.645^{o}$$

Total angle ϕ' for a shifted pin is equal to $\Rightarrow \quad \phi' = \phi + \theta = 32.645^{o}$
So the distance changed resulting from this shifting error is

$$\Rightarrow \quad I_x = l'_{pseudo} \, Cos\theta - l'_{shifted} \, Cos\phi = 6.5 \, Cos30^{o} - 6.5 \, Cos32.645^{o}$$
$$\Rightarrow \quad I_x = 0.1559mm$$

Based on the above calculations, the precision requirement of mirror inspection for checking pin alignment of the *DIL* is *±0.1559mm* tolerance in *X* direction. Since the camera resolution can achieve *0.1237mm/pixel* (Appendix K), the shifting errors in the *X* and *Y* direction can be detected during operation.

Pin Shifted in ±Z Direction — A pin length error in *±Z* direction is a special case of pin shift in *±X* direction, because no additional transfer angle ϕ exists. This error generally occurs if there is a shorter or longer pin than the correct pin length. In this case, a mirror inspection can also be applied to inspect for pin error in the *Z* direction. Although incorrect pin length is not an important factor of chip assembly, the precision requirement can be obtained from the assumption of maximum shifted distance in *X* direction. The calculation is as follows:

The angle θ between image surface and the pin image has been acquired from the above calculation. The maximum shifted distance in ±X direction is 0.3mm. Thus

$$l \, Cos\theta = 0.3 \, Cos30^{o} = 0.259mm = \text{max. acceptable tolerance}$$

So the maximum acceptable tolerance in *±Z* direction is *0.259mm*. Comparing the maximum tolerances in *X*, *Y*, and *Z* directions with vision accuracy, this mirror inspection method can reliably detect the pin alignment error of a *DIL*.

Appendix K *Camera Resolution*

Optimum resolution depends on the desired measurement of focal length, the width and height of the camera field-of-view, and the distance from the work surface to the camera. The following simple formula is based on the height of field-of-view [24]. It is useful on a camera lens selection.

$$f = S(C \div H) \qquad \text{---------- (K-1)}$$

where:

f : lens focal length in millimeter
S : camera scale factor
C : camera height
H : height of camera field-of-view

When choosing the size of the field-of-view, there is always a trade-off between image size and image resolution. When the field-of-view is large, more objects or features can be captured in each picture, which reduces the number of pictures required for the application. However, image resolution is reduced as the field-of-view is increased. On the other hand, vision processing time increases as resolution is increased. The following formulas show the relationship between vertical resolution, horizontal resolution, and field-of-view:

$$R_h = (H \div 485) \qquad \text{---------- (K-2)}$$

$$R_w = (W \div 501) \qquad \text{---------- (K-3)}$$

where :

R_h = horizontal resolution (width of one pixel)
R_w = vertical resolution (height of one pixel)
H = height of field-of-view
W = width of field-of-view

For example, a medium-resolution camera, *GP-CD 40*, has been used for chip assembly. The field-of-view acquired from the camera mounted altitude, *350mm*, is $62 \times 58mm^2$. Also, a *25mm* lens is chosen for this application. The desired resolution can be calculated from the following:

$$R_h = (H \div 485) = (58mm \div 484pixel) = 0.1198\ mm/pixel$$

$$R_w = (W \div 501) = (62mm \div 501pixel) = 0.1237\ mm/pixel$$

In this example, the resolution of *0.1237mm/pixel* is obtained. The resolution can be improved by reducing the value of H and W, which means that the camera is to be mounted lower than the current position to provide better resolution. After the resolution has been decided, the width and height of a field-of-view can be used to calculate the smallest detectable angle. Since each pixel is either black or white: by comparing the threshold value, the vision system can obtain ± 1 pixel accuracy. The dimension of pixels on the image surface is *0.1198mm* in width and *0.1237mm* in length. So the smallest detectable angle can be calculated from *Figure K-1*:

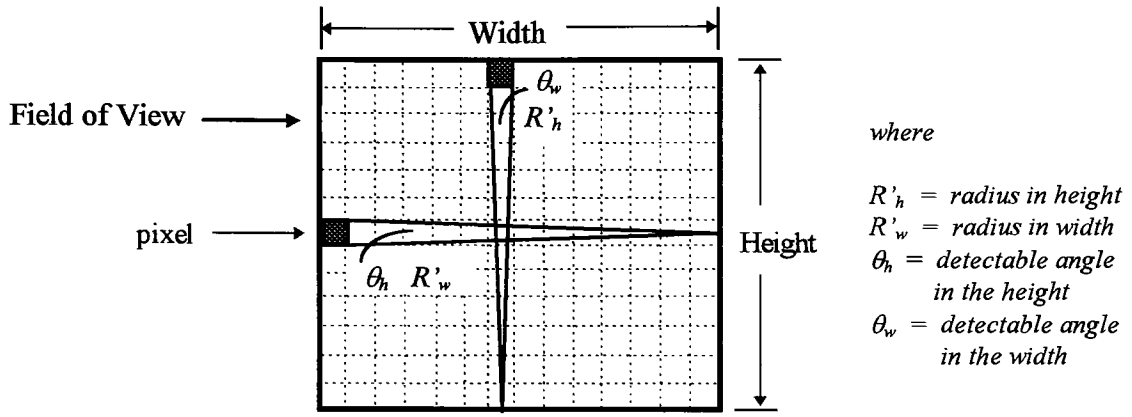


Figure K-1. Rotation Angle Determination

$$R_h = R'_w\,\theta_h \Rightarrow \theta_h = R_h / R'_w = 0.1198/62 = 1.93 \times 10^{-3}\ rad.$$

$$R_w = R'_h\,\theta_w \Rightarrow \theta_w = R_w / R'_h = 0.1237/58 = 2.13 \times 10^{-3}\ rad.$$

$$\theta_h = 1.93 \times 10^{-3}\ rad. = 0.11^{o}$$

$$\theta_w = 2.13 \times 10^{-3}\ rad. = 0.122^{o}$$

The smallest detectable angle of the vision system is 0.122^{o} in the field-of-view. Therefore, the accuracy of the rotation determination is $\pm 0.122^{o}$.

Appendix L *Assembly Program*

This program is used to control assembly flow. There are two DILs and one SMD assembly in each cycle. The location databases of DIL and SMD are named dil_*_* and smd_*_*, respectively. Digital I/O signal 12 is used to control the conveyor, ±9 control the lighting, and ±10 control solenoid actuation. Four digits I/O signals are used for vision operations. Some default commands are integrated in this program and explain as follows:

IO: digital signal output
MOVE: move robot's end effector to a location
MOVE_INSPECT: move to a specified location executing a vision inspection
OK_SIGNAL: send out a digital signal after passing an inspection
SET_FRAME: set a reference frame to the world reference frame or a vision reference frame
VISION_FRAME: a vision reference frame created by various vision tools

```
 1.  IO OUTPUT 12
 2.  MOVE TO assembly_safe
 3.  MOVE TO assembly_safe1
 4.  IO OUTPUT 9
 5.  SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pick_find1 OK_SIGNAL 2010
 6.  IF 2010
 7.    WAIT_UNTIL 1002
 8.    IF 1002
 9.      MOVE TO dil_pick_above1
10.      IO OUTPUT -9
11.      MOVE TO dil_pick_loc1
12.      IO 10
13.      IF 1001
14.        MOVE TO dil_pick_loc1
15.      END
16.      MOVE TO dil_pick_above1
17.    END
18.  END
19.  MOVE TO dil_mirr_pt1
20.  MOVE_INSPECT INSPECT dil_fing_valu1 OK_SIGNAL 2030
21.  MOVE TO dil_mirr_pt2
22.  MOVE TO dil_mirr_pt3
23.  MOVE_INSPECT INSPECT dil_fing_valu2 OK_SIGNAL 2040
24.  IF 2030 AND 2040
25.    MOVE TO assembly_safe1
26.    SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pcb_fram1 OK_SIGNAL
       2050
27.    IF 2050
28.      MOVE TO dil_place_abov1
```

```
29.       MOVE TO dil_place_loc1
30.       Next_LOOP
31.         IF 1001
32.         SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pcb_fram1
33.         MOVE TO dil_place_abov
34.         MOVE TO dil_place_loc
35.           IF 1001
36.             IO -10
37.             MOVE TO dil_place_loc
38.             MOVE TO dil_place_abov
39.             EXIT_LOOP
40.           END
41.         END
42.     ELSE
43.       MOVE TO dil_pick_above1
44.       MOVE TO dil_pick_loc1
45.       IO -10
46.     END
47.   ELSE
48.     MOVE TO dil_pick_above1
49.     MOVE TO dil_pick_loc1
50.     IO -10
51.   END
52.   MOVE TO assembly_safe1
53.   SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pick_find2
54.   OK_SIGNAL 2060
55.   IF 2060
56.     MOVE TO dil_pick_above2
57.     MOVE TO dil_pick_loc2
58.     IO 10
59.     IF 1001
60.       MOVE TO dil_pick_loc2
61.       MOVE TO dil_pick_above2
62.     END
63.   END
64.   MOVE TO dil_mirr_pt1
65.   MOVE_INSPECT INSPECT dil_fing_valu1 OK_SIGNAL 2070
66.   MOVE TO dil_mirr_pt2
67.   MOVE TO dil_mirr_pt3
68.   MOVE_INSPECT INSPECT dil_fing_valu2 OK_SINGAL 2080
69.   IF 2070 AND 2800
70.     MOVE TO assembly_safe1
71.     SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pcb_fram2
72.     OK_SGINAL 2090
73.     IF 2090
74.       MOVE TO dil_place_abov2
75.       MOVE TO dil_place_loc2
76.       IF 1001
77.       Next_LOOP 5
78.         IF 1001
79.         SET_FRAME dil_ref_frame EQUAL_TO VISION_FRAME dil_pcb_fram1
80.         MOVE TO dil_place_abov
81.         MOVE TO dil_place_loc
82.           IF 1001
83.             IO -10
```

```
84.            MOVE TO dil_place_loc
85.            MOVE TO dil_place_abov
86.            EXIT_LOOP
87.          END
88.      END
89.    ELSE
90.      MOVE TO dil_pick_above2
91.      MVOE TO dil_pick_loc2
92.      IO -10
93.    END
94.  ELSE
95.    MOVE TO dil_pick_above2
96.    MVOE TO dil_pick_loc2
97.    IO -10
98.  END
99.  MOVE TO assembly_safe1
100. SET_FRAME smd_ref_frame EQUAL_TO VISION_FRAME smd_pick_frame
101. OK_SIGNAL 2100
102. IF 2100
103.   MOVE TO smd_pick_above
104.   MOVE TO smd_pick_loc
105.   IF 1001
106.     IO 10
107.     MOVE TO smd_pick_loc
108.     MVOE TO smd_pick_above
109.   END
110. END
111. MOVE TO smd_back_above
112. MOVE TO smd_back_loc
113. IO -10
114. MOVE TO smd_back_loc
115. MOVE TO smd_back_above
116. MOVE_INSPECT INSPECT smd_fing_valu1 OK_SIGNAL 2120
117. MVOE_INSPECT INSPECT smd_fing_valu2 OK_SIGNAL 2130
118. IF 2120 AND 2130
119.   MOVE TO smd_back_above
120.   MOVE TO smd_back_loc
121.   IF 1001
122.     IO 10
123.     MOVE TO smd_back_loc
124.     MVOE TO smd_back_above
125.   END
126. ELSE
127.   MOVE TO smd_back_above
128.   MOVE TO smd_back_loc
129.   IO 10
130.   MOVE TO smd_pick_above
131.   MVOE TO smd_pick_loc
132.   IO -10
133. END
134. MOVE TO assembly_safe1
135. SET_FRAME smd_ref_frame EQUAL_TO VISION_FRAME smd_pcb_frame
136. OK_SIGNAL 2140
137. IF 2140
138.   MOVE TO smd_place_above
```

```
139.    MOVE TO smd_place_loc
140.    IF 1001
141.      MOVE TO smd_place_loc
142.      MOVE TO smd_place_above
143.    END
144.   ELSE
145.    MOVE TO smd_pick_above
146.    MOVE TO smd_pick_loc
147.    IO -10
148.  END
149.   MOVE TO dil_final_loc
150.  MOVE_INSPECT INSPECT dil_final_insp OK_SIGNAL 2150
151.  IF 2150
152.    MOVE_INSPECT INSPECT smd_final_insp OK_SIGNAL 2160
153.     IF 2150 AND 2160
154.      IO 9
155.      MOVE TO assembly_safe1
156.      IO -9
157.    ELSE
158      IO -9
159.    END
160.   ELSE
161.    IO -9
162.   END
163.  STOP_ROBOT
```